ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΑΝΕΠΙΣΤΗΜΙΟΥΠΟΛΗ ΙΩΑΝΝΙΝΩΝ, ΙΩΑΝΝΙΝΑ

22 Ιουνίου 2009

ΔΙΑΧΕΙΡΙΣΗ ΟΝΟΜΑΤΟΧΩΡΟΥ ΣΕ ΟΜΟΣΠΟΝΔΙΑΚΑ ΣΥΣΤΗΜΑΤΑ ΑΡΧΕΙΩΝ

ΑΜΒΡΟΣΙΑΔΗΣ ΓΙΩΡΓΟΣ

Επιβλέπων Καθηγητής: *Σ. Αναστασιάδης*

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Πρόλογος

Σήμερα υπάρχουν συλλογές εξυπηρετητών αρχείων, οι οποίοι συνεργάζονται για να προσφέρουν κοινούς ονοματοχώρους που αποτελούνται από συστήματα αρχείων διαθέσιμα από διαφορετικούς εξυπηρετητές και συνενωμένα αναμεταξύ τους με παραπομπές. Ο ονοματοχώρος αποτελείται από εξυπηρετητές, πελάτες, κάποια εξωτερική υπηρεσία ονοματοχώρου, και στις περισσότερες περιπτώσεις δεν είναι εύκολη ούτε η διαχείρισή του, αλλά ούτε και η διατήρηση ομοιομορφίας σε ολόκληρη την έκτασή του. Στόχος της παρούσας εργασίας είναι η υλοποίηση ενός ονοματοχώρου, ο οποίος βασίζεται σε εξυπηρετητές, είναι ομοιόμορφος στο σύνολό του, εύκολος στη διαχείρισή του και ικανός να επεκταθεί για να καλύψει σε εύρος έναν ολόκληρο οργανισμό.

Αφιερωμένο στους γονείς μου και
στην "ευρύτερη" οικογένεια μου
στη Θεσσαλονίκη

Αμβροσιάδης Γιώργος
Ιωάννινα, Ιούνιος 2009

Περίληψη στα ελληνικά

Ένας από τους σημαντικότερους στόχους της έρευνας στον τομέα των συστημάτων είναι η υλοποίηση ενός συστήματος, το οποίο θα συνδυάζει τα προτερήματα ενός κεντρικοποιημένου συστήματος με τις αρετές των κατανεμημένων συστημάτων που αναπτύσσονται επί του παρόντος. Ο στόχος αυτός βρίσκεται ακόμα μακριά από την πραγματικότητα. Παρόλα αυτά, γίνονται σημαντικά βήματα στον τομέα των συστημάτων αρχείων, ώστε να ξεκινήσει μια τέτοια συγχώνευση από χαμηλό επίπεδο.

Η εργασία περιγράφει και υλοποιεί μια υπηρεσία ικανή να δημιουργήσει έναν κοινό και ομοιόμορφο ονοματοχώρο, ο οποίος διέπεται από ιεραρχική δομή. Σε συνδυασμό με άλλες λειτουργίες, μπορεί να υποστηρίζει ανεξαρτησία θέσης για τους χρήστες και τις εφαρμογές τους, ενώ εξασφαλίζει διαφάνεια για τη θέση των δεδομένων σε ένα περιβάλλον με δυνατότητες κλιμάκωσης. Τα δεδομένα αυτά, εξάγονται με τη μορφή διαμοιραζόμενων πόρων από τους διαχειριστές των συστημάτων αρχείων στα οποία ανήκουν, και συνδυάζονται αναμεταξύ τους μέσω σημείων σύνδεσης στον κοινό ονοματοχώρο.

Για να εξασφαλιστεί η ανεξαρτησία θέσης, μια σύνδεση μεταξύ διαφορετικών συστημάτων αρχείων δεν περιέχει πληροφορίες για την πραγματική θέση των ζητούμενων δεδομένων, αλλά ένα αναγνωριστικό που μπορεί να χρησιμοποιηθεί για την αναζήτηση των πληροφοριών αυτών. Η υπηρεσία που χρησιμοποιείται για την αντιστοίχιση αυτή παρέχει ένα επίπεδο διαχωρισμού των εικονικών τοποθεσιών του ομοιόμορφου ονοματοχώρου από τις πραγματικές τοποθεσίες των αρχείων, διευκολύνοντας τις υπηρεσίες διαχείρισής του.

Στην εργασία περιοριζόμαστε στην υλοποίηση ενός τέτοιου ονοματοχώρου που υποστηρίζει μόνο το πρωτόκολλο NFSv4, παρόλο που οι κινητήριες γραμμές που δίνονται θεωρούνται επαρκείς και για την υλοποίηση σε περιβάλλοντα που χρησιμοποιούν άλλα πρωτόκολλα NAS.

Λέξεις Κλειδιά: ομοσπονδία, ομοσπονδιακός, ονοματοχώρος, κατανεμημένος, NFSv4, LDAP, σύστημα αρχείων, υπηρεσία αντιστοίχισης, σύνολο αρχείων, NSDB, ριζικό σύνολο, ριζικός εξυπηρετητής.

Περίληψη στα αγγλικά

One of the most distinctive research goals in the systems field is the implementation of a system that will be capable of merging the best features of centralized systems with those of distributed personalized systems that are currently under development. This goal is still far from being fulfilled. However, significant steps are made in the filesystems area, so that such a merging can begin from a low level.

This paper describes and implements a service capable of instantiating a common, unified namespace, with a strictly hierarchical structure. In coordination with other mechanisms, the namespace will support position independence for users and their applications, while ensuring location transparency for the data in a scalable environment. These data are exported as shared resources from the administrators of the filesystems in which they are stored, and are combined through the use of connection points in the common namespace.

In order to ensure location transparency, a connection between distinct filesystems does not contain physical location information regarding the resource at hand. Instead, it contains a unique identifier that can be used to search for physical location information. The service used for this mapping provides a level of abstraction between the logical location of the federated namespace data, and its physical location in the filesystem where they are stored. This way, it facilitates the administration services that are frequently performed.

In this paper, we implement a federated namespace as described, but we limit its support to the NFSv4 protocol alone, even though the general outline of the system described is considered sufficient to support implementation in environments using different NAS protocols as well.

Keywords: federation, namespace, distributed, federated, NFSv4, LDAP, filesystem, mapping service, NSDB, fileset, root fileset, root server.

Πίνακας περιεχομένων

1 Εισαγωγή....1

1.1 Αντικείμενο εργασίας....1

1.2 Οργάνωση Κεφαλαίων....3

2 Σκιαγράφηση προβλήματος....5

2.1 Ομοσπονδιακά Συστήματα....5

2.2 Πρωτόκολλο Εξαγωγής Δεδομένων....8

2.3 Σύνθεση Ομοσπονδιακού Ονοματοχώρου....11

2.4 Υπηρεσίες Καταλόγου και Σχεσιακές Βάσεις Δεδομένων....13

2.5 Προσεγγίσεις Μηχανισμού Αντιστοίχισης....15

2.5.1 Πρότυπο αρχείο εξαγωγής τοποθεσιών....15

2.5.2 Χρήση Σχεσιακής Βάσης Δεδομένων....16

2.5.3 Χρήση υπηρεσίας καταλόγου....16

3 Αρχιτεκτονική Ομοσπονδίας....19

3.1 Αρχιτεκτονικό Υπόβαθρο....19

3.2 Στόχοι Υλοποίησης....21

3.3 Αρχιτεκτονικός Σχεδιασμός....22

3.3.1 Χρησιμοποιούμενες τεχνολογίες....22

3.3.2 Αρχιτεκτονικός Σχεδιασμός Υλοποίησης....24

4 Υλοποίηση Ονοματοχώρου....29

4.1 Παραμετροποίηση Πυρήνα....29

4.1.1 Αντιστοίχιση στον Εξυπηρετητή....31

4.1.2 Αντιστοίχιση στον Πελάτη....32

4.2 Δαίμονας Αντιστοίχισης....33

4.3 Δαίμονας Ριζικού Συνόλου Αρχείων....34

5 Πειραματικά Αποτελέσματα....37

5.1 Χρήση περιβάλλοντος προσομοίωσης....37

5.2 Κόστος διάσχισης ομοσπονδιακού ονοματοχώρου....39

5.2.1 Διάσχιση με έναν πελάτη....40

5.2.2 Ταυτόχρονη διάσχιση με δύο πελάτες....44

5.2.3 Συμπέρασμα πειράματος....47

5.3 Κόστος κατανεμημένης δημιουργίας λογισμικού....48

5.3.1 Δημιουργία πυρήνα από έναν πελάτη....50

5.3.2 Ταυτόχρονη δημιουργία πυρήνα από δύο πελάτες....52

5.3.3 Συμπέρασμα πειράματος....55

6 Σχετικές Εργασίες....57

6.1 Common Internet Filesystem....57

6.2 Andrew Filesystem....58

6.3 Plan 9....58

6.4 Glamour....59

6.5 Storage Resource Broker....60

6.6 Domain Name System....61
6.7 Global Namespace for files....61
6.8 Federated Filesystem....62

7 Επίλογος....63

7.1 Σύνοψη και συμπεράσματα....63
7.2 Μελλοντικές επεκτάσεις....64

8 Βιβλιογραφία....65

1 *Εισαγωγή*

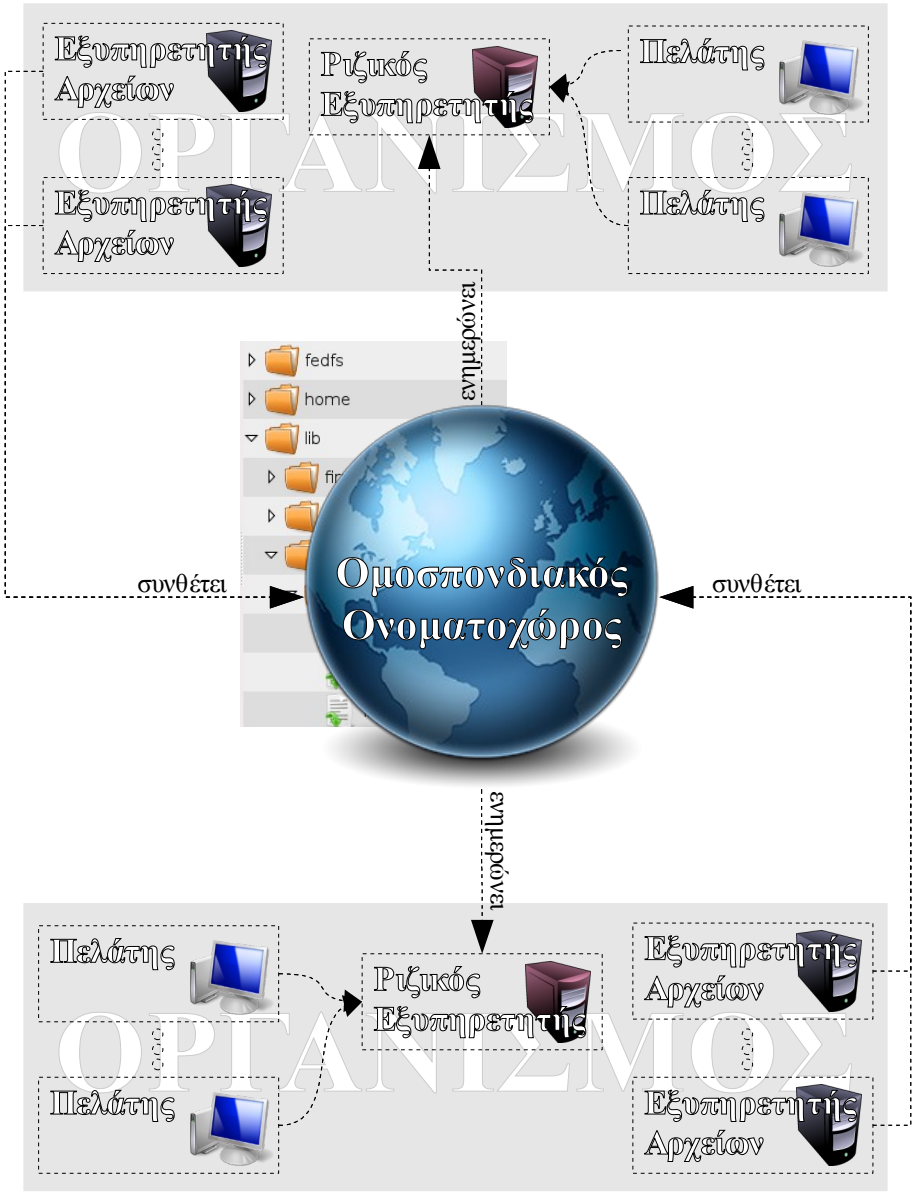
"εισαγωγή", (η) ουσ. *η ενέργεια και το αποτέλεσμα του εισάγω || το μέρος ομιλίας, βιβλίου, θεατρικού ή μουσικού έργου κλπ., που προτάσσεται*

Στο κεφάλαιο αυτό, γίνεται εισαγωγή στις βασικές έννοιες που διέπουν ένα ομοσπονδιακό σύστημα. Οι έννοιες αυτές ομαδοποιούνται στην επόμενη ενότητα, η οποία περιλαμβάνει και μια περίληψη των συμπερασμάτων που προέκυψαν κατόπιν υλοποίησης ενός ομοσπονδιακού συστήματος, και εκτέλεσης πειραμάτων σε αυτό. Τέλος, αφιερώνουμε μια ενότητα για την παρουσίαση της οργάνωσης του περιεχομένου της εργασίας στα κεφάλαια που ακολουθούν.

1.1 Αντικείμενο εργασίας

Ποιοτική περιγραφή. Ως *ομοσπονδιακό σύστημα*, ορίζουμε ένα σύστημα αρχείων, που επιτρέπει στους χρήστες του να διασχίζουν και να έχουν πρόσβαση σε δεδομένα, τα οποία διαμοιράζονται μεταξύ πολλαπλών, και ενδεχομένως απομακρυσμένων εξυπηρετητών. Οι εξυπηρετητές που εξάγουν τα δεδομένα μπορεί να ομαδοποιούνται σε *οργανισμούς*, χωρίς όμως να δίνεται η δυνατότητα σε κάποιον οργανισμό της *ομοσπονδίας* να ελέγχει τον ονοματοχώρο σε μεγαλύτερο βαθμό από τους υπόλοιπους. Ο τρόπος αποθήκευσης των δεδομένων, τα πρωτόκολλα πρόσβασης που χρησιμοποιούνται από έναν πελάτη, αλλά και ο οργανισμός στον οποίο ανήκει, δεν επηρεάζουν την ενιαία εικόνα του ονοματοχώρου που αντιλαμβάνεται. Η εικόνα του *ομοσπονδιακού ονοματοχώρου* ενδέχεται να διαφέρει μεταξύ πελατών μόνο λόγω έλλειψης επαρκών δικαιωμάτων, ή σύνδεσης σε κάποιον μη *ριζικό* ομοσπονδιακό εξυπηρετητή που δεν περιλαμβάνεται στους προκαθορισμένους για αυτή τη χρήση. Τα δικαιώματα αυτά, καθώς και η μορφή του ονοματοχώρου, καθορίζονται από

πολλαπλές διαχειριστικές οντότητες διαφορετικών οργανισμών, οι οποίες εφαρμόζουν ένα είδος κατανεμημένης διαχείρισης, που διασφαλίζει απαιτούμενη ανεξαρτησία στα δεδομένα διαφορετικών οργανισμών. Η αρχιτεκτονική ενός ομοσπονδιακού συστήματος φαίνεται στο Σχήμα 1.1, σε ποιοτική μορφή.



Σχήμα 1.1: Ποιοτική αναπαράσταση της αρχιτεκτονικής ενός ομοσπονδιακού συστήματος. Όπως φαίνεται από το σχήμα, σε κάθε οργανισμό συμμετέχει αυθαίρετος αριθμός πελατών, ριζικών, και εξυπηρετητών αρχείων. Ο ομοσπονδιακός ονοματοχώρος συντίθεται από τα δεδομένα που εξάγονται από τους εξυπηρετητές αρχείων, και γίνεται διαθέσιμος στους πελάτες μέσω των ριζικών εξυπηρετητών.

Στόχος εργασίας. Ο στόχος της εργασίας είναι να επιλύσει το πρόβλημα της δημιουργίας ενός ομοσπονδιακού ονοματοχώρου, ο οποίος προκύπτει από πολλαπλούς εξυπηρετητές, και είναι κοινός για κάθε πελάτη της ομοσπονδίας. Για την υλοποίηση χρησιμοποιείται το πρωτόκολλο NFSv4, ενώ για τη σύνθεση του ονοματοχώρου χρησιμοποιείται ειδικός

μηχανισμός, η υλοποίηση του οποίου κάνει χρήση της υπηρεσίας καταλόγου LDAP. Θέματα όπως η ασφάλεια και η επικύρωση των αναγνωριστικών κάθε χρήστη δεν εξετάζονται σε βάθος, καθώς γίνεται χρήση των υπηρεσιών που παρέχονται από τα πρωτόκολλα.

Συμπεράσματα. Για την εργασία δημιουργήθηκαν δύο υλοποιήσεις, η μια εκ των οποίων εφαρμόζει τη σύνθεση του ονοματοχώρου στους εξυπηρετητές πριν αποσταλεί η πληροφορία, ενώ η άλλη εφαρμόζεται στον εκάστοτε πελάτη, αφού η πληροφορία έχει παραλειφθεί. Τα πειράματα που εκτελέστηκαν, έδειξαν πως καμία από τις δύο υλοποιήσεις δεν υπερέχει σε ικανό επίπεδο ώστε να εκτοπίσει την άλλη. Εντούτοις, κάθε μια έχει προτερήματα αλλά και μειονεκτήματα, που την καθιστούν ιδανική για χρήση σε ορισμένες περιπτώσεις, οι οποίες θα αναλυθούν σε βάθος σε επόμενο κεφάλαιο.

1.2 Οργάνωση Κεφαλαίων

Αρχικός στόχος της εργασίας είναι να κάνει μια εισαγωγή στην έννοια του ομοσπονδιακού ονοματοχώρου. Έτσι, το **2ο κεφάλαιο** ορίζει βασικές έννοιες και παραθέτει πληροφορίες που χαρακτηρίζουν κάθε επίπεδο της αρχιτεκτονικής ενός τέτοιου συστήματος. Στο **3ο κεφάλαιο** εμβαθύνουμε περισσότερο στα ομοσπονδιακά συστήματα, θέτοντας όμως συγκεκριμένες απαιτήσεις και στόχους που αφορούν την παρούσα εργασία, ενώ γίνεται μια εισαγωγή στον αρχιτεκτονικό σχεδιασμό της υλοποίησης που δημιουργήθηκε. Η υλοποίηση και οι λεπτομέρειες που την αφορούν, όπως ενσωμάτωση χαρακτηριστικών και αντιμετώπιση προβλημάτων που προέκυψαν, περιγράφεται στο **4ο κεφάλαιο**, το οποίο επικεντρώνεται σε τεχνικά θέματα που αφορούν την εργασία και τα πρωτόκολλα που έχουν επιλεγεί. Εφόσον η περιγραφή της υλοποίησης έχει ολοκληρωθεί, στο **5ο κεφάλαιο** ακολουθεί ανάλυση των πειραμάτων που εκτελέστηκαν σε αυτή, του τρόπου διεξαγωγής και των συμπερασμάτων που προέκυψαν από τα αποτελέσματά τους.

Η μελέτη για την εργασία ήταν ευρεία, και κάλυψε συστήματα αρχείων με κατανεμημένα, αλλά και ομοσπονδιακά χαρακτηριστικά. Τα συστήματα που θεωρείται ότι πρόσκεινται στην ομοσπονδιακή προσέγγιση παρατίθενται στο **6ο κεφάλαιο**, το οποίο παρουσιάζει εκείνα τα χαρακτηριστικά τους που μας παρακίνησαν να τα μελετήσουμε, αλλά και εκείνα όπου υστερούν έναντι των ομοσπονδιακών συστημάτων. Τέλος, στο **7ο κεφάλαιο** γράφεται ο επίλογος της εργασίας, βασιζόμενος σε συμπεράσματα που προέκυψαν, καθώς και σημεία στα οποία θα μπορούσε να επικεντρωθεί μια μελλοντική έρευνα πάνω στον τομέα.

2 *Σκιαγράφηση προβλήματος*

"σκιαγραφώ", -είς, -εί ρ. *απεικονίζω αντικείμενο με τις γενικές, χαρακτηριστικές γραμμές του* || (μτφ.) *περιγράφω, αφηγούμαι κάτι σε γενικές γραμμές*

Η δημιουργία ενός ομοσπονδιακού ονοματοχώρου είναι ένα σύνθετο πρόβλημα, το οποίο προϋποθέτει την κατανόηση των αρχών που διέπουν ένα ομοσπονδιακό σύστημα σε βάθος. Το κεφάλαιο αφιερώνεται στην περιγραφή των συστατικών ενός τέτοιου συστήματος με την απαιτούμενη λεπτομέρεια, αποφεύγοντας όμως τεχνικά θέματα που αφορούν την υλοποίηση του. Οι απαιτήσεις που τίθενται εδώ έχουν αντληθεί από την βιβλιογραφία και σχετικές υλοποιήσεις, με αποτέλεσμα η τήρηση τους να κρίνεται βασική και για την υλοποίηση που παραθέτουμε.

2.1 Ομοσπονδιακά Συστήματα

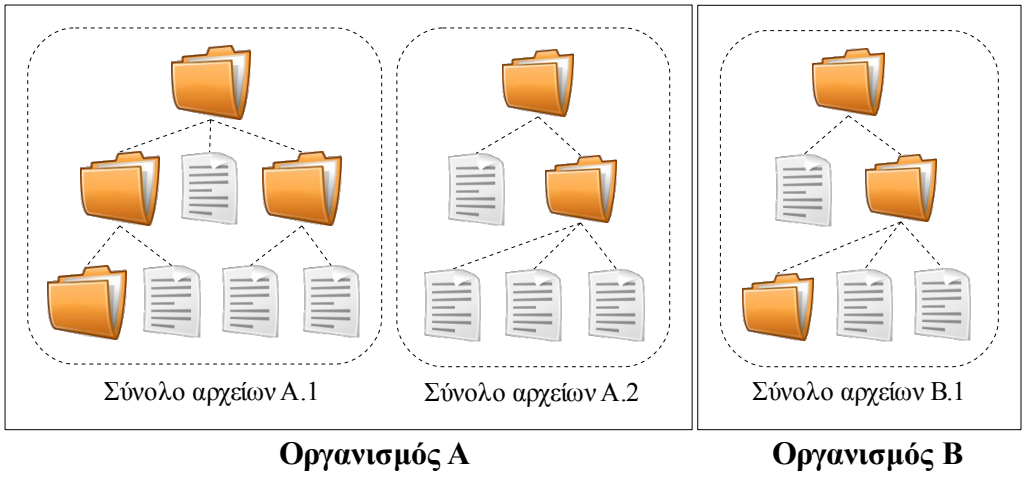
Συσχέτιση με υπάρχουσες τεχνολογίες. Αν και τα ομοσπονδιακά συστήματα εισήχθησαν μετά το 2005 στο λεξιλόγιο των επιστημόνων της πληροφορικής, εντούτοις η έννοια τους αποτελεί για πολλά χρόνια έναν από τους κυριότερους στόχους της έρευνας στον τομέα των συστημάτων. Παρόλο που υφίστανται θεωρητικές περιγραφές, αλλά και υλοποιήσεις κατανεμημένων συστημάτων από τα τέλη του 1980 [Satyanar90],[Birrell93], καμία από αυτές δεν μπορεί να θεωρηθεί ομοσπονδιακό σύστημα, κυρίως λόγω των επίπονων διαδικασιών ρύθμισης που απαιτούνται στα συστήματα εξυπηρετητών και πελατών, οι οποίες μπορεί να απαιτούν ακόμα και εγκατάσταση ειδικά προσαρμοσμένου λογισμικού στον πυρήνα του λειτουργικού συστήματος, ώστε να επιτευχθεί ομοιογένεια μεταξύ των ομοσπονδιακών κόμβων. Αντίθετα, η ρύθμιση μιας ομοσπονδίας έχει ως στόχο να είναι πολύ απλή στους

πελάτες, ανεξαρτήτως λειτουργικού συστήματος, θεωρώντας δεδομένη την ανομοιογένεια μεταξύ των κόμβων. Ωστόσο, πολλές από τις υλοποιήσεις αυτές μοιράζονται επιμέρους χαρακτηριστικά και στόχους με την δημιουργία μιας ομοσπονδίας. Για αυτό το λόγο, παρουσιάζονται στο Κεφάλαιο 6 ως μέτρο σύγκρισης.

Αρχιτεκτονική ομοσπονδιακού συστήματος. Ξεφεύγοντας από την ποιοτική θεώρηση, ένα ομοσπονδιακό σύστημα μπορούμε να πούμε ότι αποτελεί τον συνδετικό κρίκο μεταξύ ετερογενών συστημάτων αρχείων που εξάγονται από τους εξυπηρετητές της ομοσπονδίας. Τα δεδομένα αυτά συντίθενται σε έναν ενιαίο *ομοσπονδιακό ονοματοχώρο*, ο οποίος διατίθεται έπειτα στους πελάτες. Όμως ο στόχος του πρωτοκόλλου της ομοσπονδίας δεν περιορίζεται εκεί. Για την επίτευξη πραγματικής ανεξαρτησίας των δεδομένων, αλλά και ευελιξίας στη διαχείριση τους μεταξύ οργανισμών, εισάγουμε ένα επίπεδο αφαίρεσης στον καθορισμό της τοποθεσίας τους. Μέσω του *μηχανισμού αντιστοίχισης*, οι πελάτες αντιλαμβάνονται τα δεδομένα σύμφωνα με την εικονική τους θέση στον ονοματοχώρο της ομοσπονδίας. Η τοποθεσία αυτή συνδυάζεται έπειτα, μέσω του μηχανισμού, με την πραγματική τοποθεσία σε κάποιο εξαγόμενο φάκελο ενός εξυπηρετητή. Ο μηχανισμός αντιστοίχισης καθίσταται ιδιαίτερα χρήσιμος, καθώς παρέχει διαφάνεια στους πελάτες της ομοσπονδίας ως προς την πραγματική θέση των δεδομένων. Αυτό συνεπάγεται την εύκολη μεταβολή της τελευταίας είτε κατά τη χρήση μηχανισμών αναπαραγωγής (*replication*), που παραπέμπουν τον πελάτη σε ακριβές αντίγραφο των δεδομένων σε διαφορετική τοποθεσία, είτε λόγω μετακίνησης (*migration*) των δεδομένων σε νέα τοποθεσία. Στα επόμενα κεφάλαια θα εξεταστούν υπάρχουσες τεχνολογίες που μπορούν να εφαρμοστούν στα επιμέρους τμήματα της ομοσπονδίας, ενώ ο μηχανισμός αντιστοίχισης, ο οποίος αποτελεί και το κύριο θέμα της εργασίας θα εξεταστεί με περισσότερη λεπτομέρεια, παραθέτοντας και τρόπους υλοποίησής του.

Σύνολα αρχείων. Κάθε οργανισμός της ομοσπονδίας, εξάγει μέσω των εξυπηρετητών του ένα μέρος του ομοσπονδιακού ονοματοχώρου, στον οποίο στη συνέχεια έχουν πρόσβαση όλοι οι εξυπηρετητές, και κατά επέκταση οι πελάτες της ομοσπονδίας. Τα δεδομένα στον ομοσπονδιακό ονοματοχώρο είναι δομημένα ιεραρχικά, όπως και στον ονοματοχώρο ενός τοπικού συστήματος αρχείων. Ο τρόπος με τον οποίο συντίθενται τα εξαγώγιμα δεδομένα ώστε να προκύψει τελικά ο ομοσπονδιακός ονοματοχώρος θα γίνει εμφανής σε επόμενη ενότητα. Τα τμήματα ή *σύνολα αρχείων*, στα οποία διαμερίζεται, είναι απλές ομάδες αρχείων και φακέλων, και μπορεί να περιέχουν από ένα αρχείο μέχρι ένα ολόκληρο σύστημα αρχείων, χωρίς όμως να ξεφεύγουν από τα όρια του συστήματος αρχείων όπου ανήκουν. Όπως φαίνεται και στο Σχήμα 2.1, ένας οργανισμός δεν είναι απαραίτητο να εξάγει τα δεδομένα του σε ιεραρχική μορφή, καθώς η σύνθεση του ονοματοχώρου της ομοσπονδίας γίνεται σε

επόμενο στάδιο. Ένα σύνολο αρχείων μπορεί να ανήκει μόνο σε ένα σύστημα αρχείων ενός εξυπηρετητή, όμως δεν ισχύει και το αντίστροφο. Οι πελάτες ενός οργανισμού μπορούν να συνδεθούν με οποιονδήποτε εξυπηρετητή και να προσπελάσουν τα δεδομένα που εξάγει, όμως θεωρούμε πως η πρόσβαση στη ρίζα του ομοσπονδιακού ονοματοχώρου μπορεί να επιτευχθεί μόνο με σύνδεση στους *ριζικούς εξυπηρετητές* της ομοσπονδίας.



Σχήμα 2.1: Τα σύνολα αρχείων των οργανισμών δεν απαιτείται να διέπονται από ιεραρχία. Κάθε οργανισμός μπορεί να εξάγει αυθαίρετο αριθμό συνόλων αρχείων, κάθε ένα από τα οποία μπορεί να είναι ανεξάρτητο από τα υπόλοιπα. Μόνη απαίτηση, και προφανής, είναι τα σύνολα αρχείων να διέπονται από ιεραρχία, ώστε να μπορούν να χρησιμοποιηθούν αυτούσια ως κομμάτι του ονοματοχώρου. Σε διαφορετική περίπτωση, το σύνολο προς εξέταση μπορεί να διαχωριστεί σε περισσότερα από ένα και να διαχωριστεί σε τμήματα, όπως στο σχήμα.

Ομοσπονδιακός ονοματοχώρος. Ο ομοσπονδιακός ονοματοχώρος είναι ομοιόμορφος, δηλαδή έχει τη δυνατότητα να διαθέτει τα διαμοιραζόμενα δεδομένα σε κάθε πελάτη μέσω των ίδιων τοποθεσιών, παρόμοια με ένα τοπικό σύστημα αρχείων με πολλούς χρήστες. Οι ρυθμίσεις προσάρτησης του ονοματοχώρου που γίνονται από τον πελάτη είναι σαφείς και στατικές. Αυτό σημαίνει ότι πιθανές αλλαγές στον ομοσπονδιακό ονοματοχώρο δεν απαιτούν την αλλαγή τους. Τα σύνολα αρχείων που αποτελούν τον ονοματοχώρο, γίνονται διαθέσιμα σε διάφορα σημεία του μέσω απλών *προσαρτήσεων* (*mount*), δηλαδή συνδέσεων της ρίζας του συνόλου αρχείων σε κάποιον φάκελο του υπάρχοντος ονοματοχώρου. Εν ολίγοις, ο ομοσπονδιακός ονοματοχώρος αποτελείται από την σύνθεση πολλαπλών συνόλων αρχείων με προσαρτήσεις, ενώ οι φυσικές τοποθεσίες των συνόλων αρχείων μπορούν να βρίσκονται σε διαφορετικούς εξυπηρετητές. Η ομοσπονδία δεν απαιτεί ο ονοματοχώρος να είναι κοινός για κάθε πελάτη, καθώς υπάρχει η δυνατότητα να δημιουργηθούν ανεξάρτητοι ονοματοχώροι, που επιτρέπουν την μεταξύ τους διάσχιση μέσω ειδικά ορισμένων τοπικών σημείων προσάρτησης που καθορίζονται από τον χρήστη, αν και κάτι τέτοιο αναιρεί την έννοια της ομοσπονδίας, καθώς διασπά την ομοιομορφία του ομοσπονδιακού ονοματοχώρου.

Σημεία προσάρτησης και αναφορές. Η σύνθεση του ομοσπονδιακού ονοματοχώρου από σύνολα αρχείων, γίνεται μέσω της εισαγωγής συνδετικών σημείων προσάρτησης. Ως σημείο

προσάρτησης ορίζεται ένας κατάλογος σε κάποιο πατρικό σύνολο αρχείων, στον οποίο συνδέεται ένα άλλο (θυγατρικό) σύνολο. Όταν συναντάται το σημείο προσάρτησης ενός θυγατρικού συνόλου αρχείων, τότε η διάσχιση του συνεχίζει διαφανώς από τη ρίζα του συνόλου αυτού, δεδομένου ότι διαθέτει τα απαραίτητα δικαιώματα. Συγκεκριμένα, όταν ένας πελάτης διασχίζει ένα σημείο πρόσβασης, τότε παραπέμπεται σε μια λίστα τοποθεσιών, στις οποίες βρίσκονται τα δεδομένα του ζητούμενου συνόλου αρχείων. Ο πελάτης έχει έπειτα τη δυνατότητα να επιλέξει μια εξ αυτών και να συνδεθεί εκεί. Η λειτουργία που περιγράψαμε ονομάζεται αναφορά (*referral*). Τέλος, ένα σύνολο αρχείων μπορεί να προσαρτάται περισσότερες από μια φορές στον ονοματοχώρο, ακόμα και σε εμφωλευμένη μορφή.

Ριζικοί εξυπηρετητές και Ριζικό σύνολο αρχείων. Όπως διατυπώθηκε προηγουμένως, οι πελάτες της ομοσπονδίας μπορούν να συνδεθούν σε οποιονδήποτε ομοσπονδιακό εξυπηρετητή, αρκεί να έχουν επαρκή δικαιώματα, όμως πρόσβαση στον ομοσπονδιακό ονοματοχώρο παρέχει μόνο ένα σύνολο υποδεδειγμένων *ριζικών εξυπηρετητών*. Το σύνολο αρχείων που εξάγεται από αυτούς ονομάζεται *ριζικό σύνολο αρχείων*, αποτελεί τη ρίζα του ομοσπονδιακού ονοματοχώρου και είναι κοινό, δηλαδή αναπαράγεται μέσω ενημερώσεων σε κάθε ριζικό εξυπηρετητή. Θεωρούμε πως σε κάθε ομοσπονδία υφίσταται ένας τουλάχιστον τέτοιος εξυπηρετητής, αν και δίνεται πάντοτε η δυνατότητα στον πελάτη να συνδεθεί σε εξυπηρετητή διαφορετικής ομοσπονδίας, εφόσον διαθέτει τα απαραίτητα δικαιώματα. Παρόλα αυτά, δεν απαιτείται για κανέναν πελάτη να προσαρτήσει τον ονοματοχώρο από τους ριζικούς εξυπηρετητές. Στην περίπτωση που χρησιμοποιηθεί κάποιος από τους υπόλοιπους, ο πελάτης θα μπορεί να διασχίσει μόνο το κομμάτι του ονοματοχώρου που ορίζεται από εκείνον. Επίσης, μπορεί να προσαρτήσει τοπικά πολλαπλά διαφορετικά σύνολα αρχείων από διαφορετικούς εξυπηρετητές, και να τα συνθέσει ο ίδιος, αν και όπως διατυπώθηκε, αυτό αναιρεί την έννοια της ομοσπονδίας.

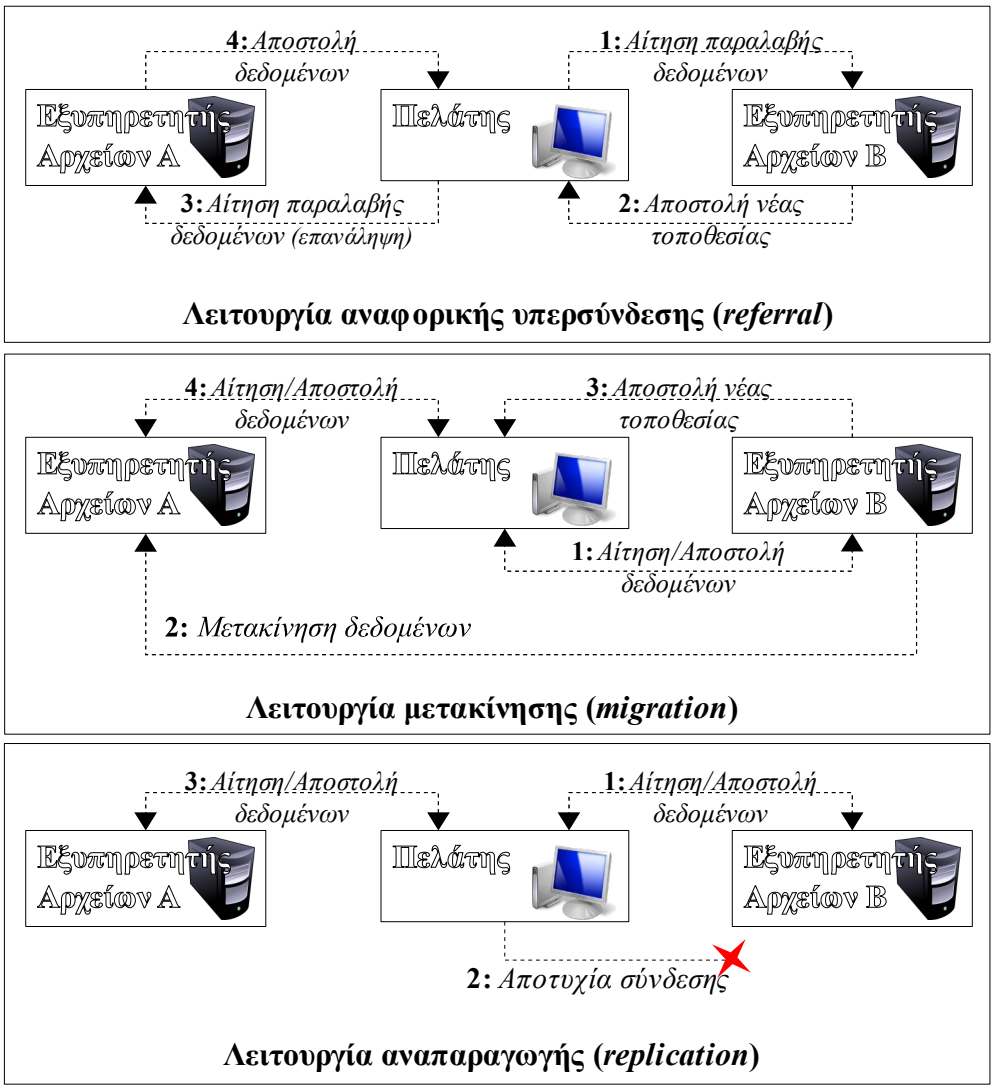
2.2 Πρωτόκολλο Εξαγωγής Δεδομένων

Το πρώτο βήμα στη δημιουργία ενός ομοσπονδιακού συστήματος, είναι να δοθεί η δυνατότητα στους ομοσπονδιακούς εξυπηρετητές να εξάγουν σύνολα δεδομένων στην ομοσπονδία. Τα *πρωτόκολλα εξαγωγής δεδομένων* που μπορούν να χρησιμοποιηθούν για την εξαγωγή των συνόλων αρχείων πρέπει να υποστηρίζουν συγκεκριμένα χαρακτηριστικά, τα οποία κρίνονται απαραίτητα για την εύρυθμη λειτουργία ενός ομοσπονδιακού συστήματος.

Κλείδωμα αρχείων. Η δυνατότητα κλειδώματος αρχείων (*locking*) κατά την προσπέλαση τους από πελάτες θεωρείται δεδομένο χαρακτηριστικό για τα πρωτόκολλα επικοινωνίας που χρησιμοποιούνται στη δημιουργία μιας ομοσπονδίας. Χωρίς αυτή, η διατήρηση της συνέπειας

των δεδομένων κατά την μεταβολή τους από κάποιον πελάτη φαίνεται αδύνατη. Η υποστήριξή της, όμως, είναι διαδεδομένη σε όλα τα σύγχρονα πρωτόκολλα κατανεμημένων συστημάτων, επομένως δεν δημιουργεί πρόβλημα στην επιλογή πρωτοκόλλου.

Προσωρινή αποθήκευση. Μια ομοσπονδία οφείλει να μπορεί να εγγυηθεί στους πελάτες της όσο το δυνατόν ταχύτερη παροχή δεδομένων κατά τη χρήση του ονοματοχώρου της, και αυτό αποκλείει την ύπαρξη καθυστερήσεων κατά τις μεταβάσεις μεταξύ συνόλων αρχείων. Η λειτουργία προσωρινής κρυφής αποθήκευσης (*caching*) που είναι διαθέσιμη στα περισσότερα πρωτόκολλα μπορεί να εγγυηθεί αυξημένη ταχύτητα προσπέλασης που καλύπτει σε μεγάλο ποσοστό τον χρόνο διάσχισης ενός σημείου προσάρτησης, κάνοντας έτσι σχεδόν διαφανή στους πελάτες της ομοσπονδίας την μετακίνηση τους αναμεταξύ συνόλων αρχείων.



Σχήμα 2.2: Στην περίπτωση της αναφορικής υπερσύνδεσης η πληροφορία βρίσκεται στον εξυπηρετητή Α, και ο χρήστης ενημερώνεται για την τοποθεσία των δεδομένων ώστε να ανακατευθύνει την αίτηση του εκεί. Στην περίπτωση μετακίνησης των δεδομένων, η ενημέρωση για τη νέα θέση γίνεται κατά την αίτηση για αυτά στον πρώην εξυπηρετητή. Τέλος, στην περίπτωση αναπαραγωγής ο πελάτης έχει πρότερη γνώση των αντιγράφων, και αν κάποια αίτηση του προς ένα αντίγραφο αποτύχει, δοκιμάζει με το επόμενο.

Αναφορικές υπερσυνδέσεις. Ένα ιδιαίτερα σημαντικό χαρακτηριστικό, το οποίο οφείλουν να διαθέτουν τα πρωτόκολλα που πρόκειται να χρησιμοποιηθούν για τη δημιουργία ομοσπονδιακών συστημάτων, είναι οι αναφορικές υπερσυνδέσεις (*referrals*). Με αυτή τη λειτουργία επιτυγχάνονται συνδέσεις μεταξύ συνόλων αρχείων. Συγκεκριμένα, ένα σύνολο αρχείων μπορεί να συνδεθεί με ένα άλλο, προσαρτώντας το με αναφορά. Έτσι, κάθε φορά που ένας πελάτης διασχίζει το σημείο προσάρτησης, το πρωτόκολλο φροντίζει να τον ενημερώνει για τη θέση του νέου συνόλου, και η διάσχιση συνεχίζεται από τη νέα ρίζα, που προσαρτάται στο τρέχον μονοπάτι διάσχισης. Σχηματική αναπαράσταση της λειτουργίας αναφορικής υπερσύνδεσης φαίνεται στο Σχήμα 2.2.

Απόδοση σε δίκτυα ευρείας περιοχής. Όπως θα δούμε και παρακάτω, τα πρωτόκολλα που προτιμώνται σήμερα σε περιβάλλοντα παραγωγής είναι σχεδιασμένα για δίκτυα, οι κόμβοι των οποίων δεν έχουν μεγάλη γεωγραφική απόσταση μεταξύ τους. Τα πρωτόκολλα που έχουν σχεδιαστεί για χρήση σε τοπικά δίκτυα λαμβάνουν υπ' όψιν τους στη σύνθεση των αιτήσεων που χρησιμοποιούν, τους μικρούς χρόνους απόκρισης των τοπικών δικτύων, έχοντας ως αποτέλεσμα να δημιουργούνται προβλήματα κατά τη χρήση τους σε δίκτυα ευρείας περιοχής, όπου οι χρόνοι καθυστέρησης που παρατηρούνται είναι έντονοι. Η απαίτηση των ομοσπονδιακών συστημάτων να υποστηρίζονται πολλαπλοί, γεωγραφικά απομακρυσμένοι οργανισμοί, μας αναγκάζει να απορρίψουμε τα πρωτόκολλα αυτά, και να στραφούμε σε νεότερες τεχνολογίες, όπως το πρωτόκολλο NFSv4 που επιλέγεται για την υλοποίηση.

Ασφάλεια. Όπως αναμένεται, αυξάνοντας την κλίμακα του δικτύου όπου χρησιμοποιείται ένα πρωτόκολλο, αυξάνεται και η απαίτηση να είναι ασφαλές. Πρωτόκολλα που, όπως αναφέρθηκε πριν, έχουν σχεδιαστεί για χρήση σε δίκτυα μικρής κλίμακας, έχουν σημαντικό πρόβλημα κλιμάκωσης σε μεγαλύτερα περιβάλλοντα από άποψη ασφάλειας. Λόγω του γεγονότος πως το περιβάλλον στο οποίο καλείται να λειτουργήσει ένα ομοσπονδιακό σύστημα είναι ιδιαίτερα εχθρικό καθώς μπορεί να εκτείνεται μεταξύ πόλεων, χωρών ή ακόμα και ηπείρων, είναι απαραίτητο να διατίθενται ενσωματωμένοι ανεπτυγμένοι μηχανισμοί ασφάλειας.

Μετακίνηση και αναπαραγωγή. Η λειτουργία μετακίνησης (*migration*) επιτρέπει στον διαχειριστή ενός συστήματος αρχείων να αλλάξει την τοποθεσία των δεδομένων, χωρίς να δημιουργηθεί πρόβλημα στους χρήστες του συστήματος. Το πρωτόκολλο αναλαμβάνει να παραπέμψει τους χρήστες προς την νέα τοποθεσία διαφανώς, και η λειτουργία του συστήματος δεν διακόπτεται. Σχεδόν παρόμοια, μέσω της λειτουργίας αναπαραγωγής (*replication*), ο διαχειριστής του συστήματος μπορεί να δημιουργήσει περισσότερες από μια

τοποθεσίες αποθήκευσης των δεδομένων, οι οποίες είναι μεταξύ τους πανομοιότυπα αντίγραφα. Έτσι, όταν ο εξυπηρετητής με τον οποίο συνδιαλέγεται το πρωτόκολλο πάψει να είναι διαθέσιμος, μπορεί να απευθυνθεί σε έναν από τους υπόλοιπους εξυπηρετητές που διαθέτουν αντίγραφα των δεδομένων. Η υπηρεσία αναπαραγωγής αναλαμβάνει την δυναμική ενημέρωση όλων των αντιγράφων, ώστε τα δεδομένα να είναι πάντοτε έγκυρα. Σχηματικές αναπαραστάσεις των λειτουργιών μετακίνησης και αναπαραγωγής φαίνονται στο Σχήμα 2.2.

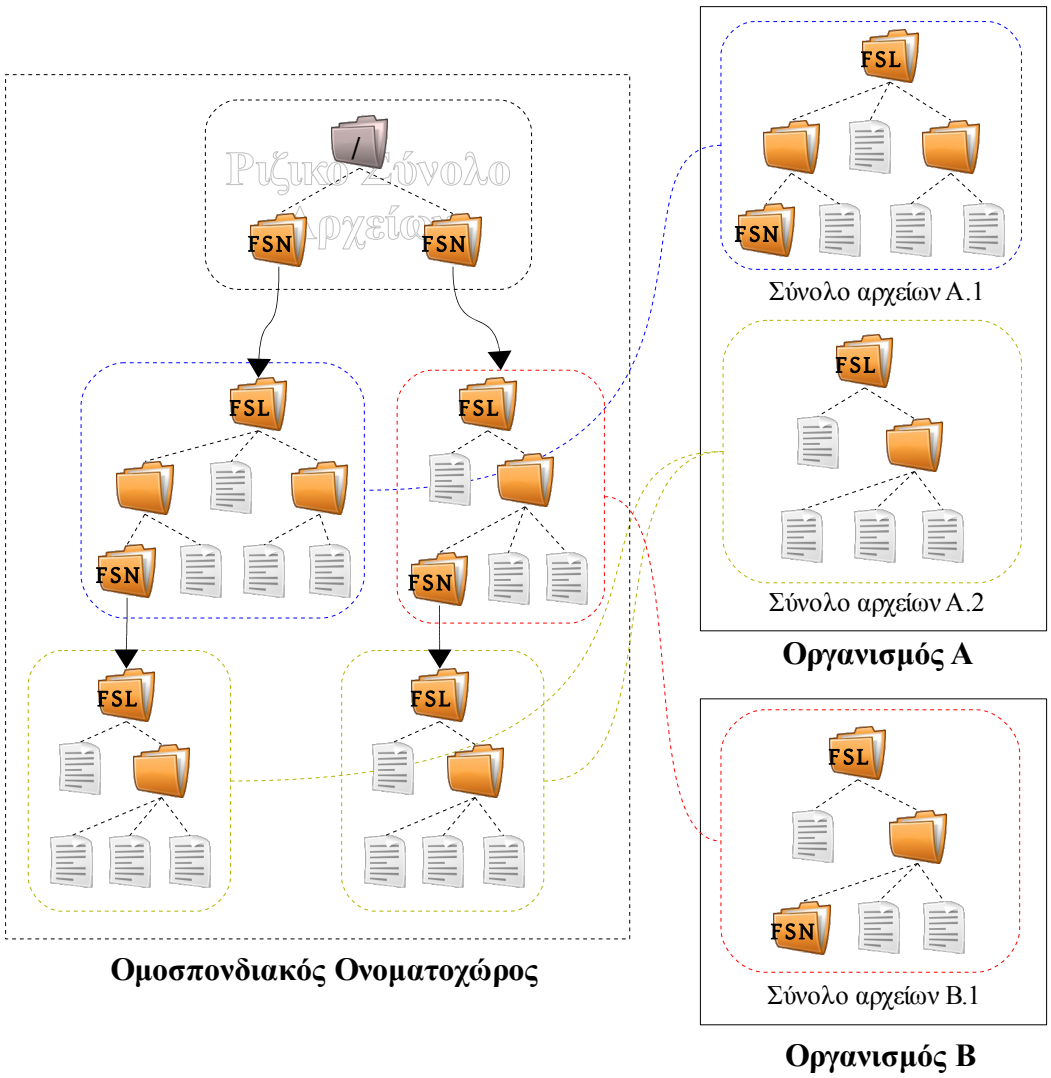
2.3 Σύνθεση Ομοσπονδιακού Ονοματοχώρου

Η ενότητα αυτή ολοκληρώνει την σκιαγράφηση της σύνθεσης ενός ομοσπονδιακού συστήματος και των συστατικών που το αποτελούν, αναλύοντας σε μεγαλύτερο βάθος τον μηχανισμό σύνθεσης του ομοσπονδιακού ονοματοχώρου και τις έννοιες που τον συνθέτουν, ολοκληρώνοντας έτσι την ανάλυση των αρχιτεκτονικών μερών που αποτελούν ένα ομοσπονδιακό σύστημα.

Μηχανισμός αντιστοίχισης. Όπως αναφέρθηκε και προηγουμένως, ένα από τα χαρακτηριστικά του πρωτοκόλλου της ομοσπονδίας είναι η ικανότητά του να διαχωρίζει την πραγματική θέση των δεδομένων, από την θέση τους στον ομοσπονδιακό ονοματοχώρο. Αυτός ο διαχωρισμός επιτυγχάνεται μέσα από ένα μηχανισμό, ο οποίος αποδίδει μοναδικά ονόματα στα σύνολα αρχείων και τις τοποθεσίες τους, και έπειτα τα αντιστοιχίζει κατά απαίτηση. Ο μηχανισμός αντιστοίχισης παρέχει ευελιξία στον ονοματοχώρο, καθώς μπορεί να εκμεταλλευτεί τις λειτουργίες αναπαραγωγής και μετακίνησης που διατίθενται από το πρωτόκολλο εξαγωγής δεδομένων, ενημερώνοντας κάθε φορά τον πελάτη για την τρέχουσα τοποθεσία μέσω της λειτουργίας αναφορικής υπερσύνδεσης. Η χρήση του μηχανισμού αυτού, απαιτείται σε κάθε διάσχιση όταν αυτή φτάσει σε σημείο προσάρτησης, όπου παρέχεται το αναγνωριστικό του θυγατρικού συνόλου αρχείων, το οποίο υποβάλλεται στον μηχανισμό για επιστροφή της τοποθεσίας. Στην περίπτωση που υποστηρίζεται αναπαραγωγή, ο μηχανισμός μπορεί να επιστρέψει περισσότερες από μία τοποθεσίες, οι οποίες περιέχουν αντίγραφα του ίδιου συνόλου αρχείων. Από τις τοποθεσίες αυτές, επιλέγεται η ιδανική και συνεχίζεται η διάσχιση του ονοματοχώρου. Στην περίπτωση μετακίνησης των δεδομένων η νέα τοποθεσία προωθείται εύκολα στους πελάτες, οι οποίοι ενημερώνονται για αυτή στην επόμενη αίτηση δεδομένων από το εν λόγω σύνολο αρχείων. Τα δεδομένα της αντιστοίχισης αποθηκεύονται σε συγκεκριμένους κόμβους της ομοσπονδίας, με αποτέλεσμα μετακινήσεις, αλλά και προσθήκες αντιγράφων συνόλων αρχείων να συνεπάγονται απλά την ενημέρωση τους.

Αναγνωριστικά συνόλων αρχείων και τοποθεσιών αποθήκευσης. Τα σύνολα αρχείων αναγνωρίζονται από το μηχανισμό αντιστοίχισης μέσω των μοναδικών ονομάτων που τους

αποδίδονται (*fileset name* ή *FSN*). Τα ονόματα αυτά θεωρούμε ότι είναι μοναδικά κατά μήκος ολόκληρης της ομοσπονδίας και συνοδεύονται από τη διεύθυνση κάποιου κόμβου στον οποίο υλοποιείται ο μηχανισμός αναζήτησης και περιέχονται οι ζητούμενες τοποθεσίες. Οι κόμβοι αυτοί ονομάζονται κόμβοι υπηρεσίας δεδομένων ονοματοχώρου, ή κόμβοι NSDB (*NSDB nodes* ή *namespace database* nodes). Όσων αφορά τα σύνολα αρχείων, θεωρούμε πως κάθε ένα από αυτά, από τη δημιουργία του, αντιστοιχίζεται με ένα μοναδικό FSN και συσχετίζεται τουλάχιστον με μια τοποθεσία αποθήκευσης σε κάποιον ομοσπονδιακό εξυπηρετητή (*fileset location* ή *FSL*). Κάθε τοποθεσία FSL αποτελείται από πολλαπλά χαρακτηριστικά, τα οποία είναι απαραίτητα για την προσάρτηση του συνόλου. Θεωρείται δεδομένο, πως η ύπαρξη περισσότερων από μιας τοποθεσιών, υπονοεί την ύπαρξη αντιγράφων του συνόλου σε διαφορετικούς εξυπηρετητές.



Σχήμα 2.3: Σχηματική απεικόνιση της σύνθεσης ενός ομοσπονδιακού ονοματοχώρου από τα εξαγόμενα σύνολα αρχείων των οργανισμών Α και Β. Η διάσχιση ξεκινά από το ριζικό σύνολο αρχείων που εξάγεται από τον ριζικό εξυπηρετητή, και κάθε φορά που ένας πελάτης μεταβαίνει σε ένα σημείο σύνδεσης ενός συνόλου αρχείων χρησιμοποιεί το μοναδικό του όνομα (FSN) για να βρει την τοποθεσία του (FSL). Να σημειωθεί, ότι ένα σύνολο αρχείων μπορεί να προσαρτηθεί οσεσδήποτε φορές στον ονοματοχώρο, ακόμα και εμφωλευμένα.

Υπηρεσία δεδομένων ονοματοχώρου. Η υπηρεσία που αναλαμβάνει την υλοποίηση και παροχή του μηχανισμού αντιστοίχισης ονομάζεται *υπηρεσία δεδομένων ονοματοχώρου*, και εκτείνεται σε ολόκληρη την ομοσπονδία. Η υπηρεσία υλοποιείται στους κόμβους NSDB που περιγράφηκαν προηγουμένως, μέσω διεπαφών οι οποίες επιτρέπουν ορισμό, ενημέρωση και αναζήτηση δεδομένων που αφορούν τα σύνολα αρχείων (FSN), καθώς και πληροφορίες για την αντιστοίχιση τους με τοποθεσίες (FSLs). Την διαχείριση των κόμβων αναλαμβάνουν διαχειριστικές οντότητες, οι οποίες μπορούν να διαχειρίζονται περισσότερους από έναν, χωρίς όμως να ισχύει και το αντίστροφο. Η διαφορά μεταξύ της υπηρεσίας NSDB και ενός κόμβου που την παρέχει, είναι ανάλογη της διαφοράς μίας υπηρεσίας DNS από έναν εξυπηρετητή DNS. Στην ομοσπονδία υφίστανται "*τοπικοί*" και "*απομακρυσμένοι*" κόμβοι NSDB, εννοώντας τους κόμβους που ανήκουν στον ίδιο ή σε διαφορετικό οργανισμό, αντίστοιχα, από το σημείο αναφοράς. Οι τοπικοί κόμβοι NSDB, σύμφωνα με το [FedFS-Reqs] μπορούν να θεωρηθούν γνωστοί χωρίς σχετική αναζήτηση στον εξυπηρετητή. Οι ομοσπονδιακοί εξυπηρετητές μπορούν πάντοτε να επικοινωνούν είτε με κάποιον τοπικό NSDB κόμβο (αν έχει οριστεί ένας τέτοιος), είτε με οποιονδήποτε άλλο NSDB κόμβο, για να επιλύσουν την αντιστοίχιση κάποιας ένωσης. Η ύπαρξη τοπικού εξυπηρετητή σε έναν οργανισμό είναι προαιρετική, ενώ η αναπαραγωγή τους σε αντίγραφα είναι αποδεκτή, αν χρησιμοποιηθούν οι κατάλληλες διεπαφές. Γενικά, θεωρούμε ότι η επανάληψη αυτού του είδους πληροφορίας δεν θεωρείται ότι δημιουργεί πρόβλημα στην ομοσπονδία. Οι διεπαφές που μπορούν να χρησιμοποιηθούν για τη λειτουργία ενός κόμβου NSDB είναι δυνατό να υλοποιούν από μια απλή υπηρεσία δημιουργίας στατικών αρχείων ρυθμίσεων, μέχρι ένα σύστημα διαχείρισης κάποιας σχεσιακής βάσης δεδομένων, ή υπηρεσίας καταλόγου όπως το LDAP.

2.4 Υπηρεσίες Καταλόγου και Σχεσιακές Βάσεις Δεδομένων

Σε αυτό το σημείο, θα ήταν χρήσιμο να εξηγήσουμε τα χαρακτηριστικά μιας υπηρεσίας καταλόγου σε αντιδιαστολή με αυτά μιας σχεσιακής βάσης δεδομένων γενικής χρήσης, ώστε να γίνουν αντιληπτοί οι λόγοι για τους οποίους μπορεί να επιλεγεί μια από αυτές τις διεπαφές έναντι της άλλης, για τη δημιουργία ενός κόμβου δεδομένων ονοματοχώρου.

Υπηρεσίες καταλόγου. Ο κατάλογος είναι μία δομή πληροφοριών, η οποία αποτελείται από οργανωμένα αντικείμενα για τα οποία διατηρούνται πληροφορίες. Ένα συνηθισμένο παράδειγμα υπηρεσίας καταλόγου είναι ο τηλεφωνικός κατάλογος. Ένας κατάλογος μπορεί να οριστεί ως μια εξειδικευμένη βάση δεδομένων, η οποία εξυπηρετεί σχεδόν στατικές εγγραφές, ενώ παράλληλα είναι προσαρμοσμένη να χειρίζεται με αποδοτικό τρόπο ερωτήματα αναζήτησης που συνεπάγονται μεγάλο όγκο πληροφορίας. Οι κατάλογοι κάνουν

χρήση ενός βελτιστοποιημένου πρωτοκόλλου υποβολής επερωτήσεων, που μπορεί να χρησιμοποιηθεί σε μη απαιτητικές εφαρμογές. Λόγω του γεγονότος πως διαχειρίζονται κυρίως αιτήματα ανάγνωσης, κάποιες από τις υλοποιήσεις καταλόγων δεν υποστηρίζουν συναλλαγές. Οι υλοποιήσεις αυτές δεν θα ήταν πρακτικές στο ομοσπονδιακό περιβάλλον και έτσι συνήθως χρησιμοποιούνται υβριδικές υλοποιήσεις, οι οποίες περιέχουν τις απαραίτητες λειτουργίες. Σημαντικά χαρακτηριστικά υλοποιήσεων υπηρεσιών καταλόγου αποτελούν οι ενσωματωμένοι μηχανισμοί ασφάλειας, η υποστήριξη αναπαραγωγής (*replication*) και μετακίνησης (*migration*), καθώς και η ιεραρχική διαχείριση των δεδομένων. Στην ιεραρχική διαχείριση, τα δεδομένα διατάσσονται σε ιεραρχική μορφή στη βάση. Αυτό δημιουργεί πρόβλημα στην τροποποίηση των εγγραφών, καθώς λειτουργίες όπως η μετονομασία επηρεάζουν όλες τις εξαρτώμενες εγγραφές, λόγω της ιεραρχικής τους δόμησης. Συνήθως το μοντέλο ονομασίας κάνει χρήση ενός μοναδικού χαρακτηριστικού που προσδιορίζει την εγγραφή και εξαρτάται από το αντίστοιχο χαρακτηριστικό της πατρικής εγγραφής. Με το μηχανισμό αυτό, επιτυγχάνεται ιεράρχηση, αλλά επιβάλλεται η τροποποίηση κάθε γονικής ή εξαρτώμενης εγγραφής στην περίπτωση που το εν λόγω χαρακτηριστικό τροποποιηθεί.

Σχεσιακές Βάσεις Δεδομένων. Σε αντίθεση με τις υπηρεσίες καταλόγου, οι σχεσιακές βάσεις δεδομένων γενικής χρήσης είναι περισσότερο αποδοτικές για την αποθήκευση δεδομένων τα οποία αλλάζουν συχνά. Επίσης, παρέχουν τη δυνατότητα τροποποίησης μιας εγγραφής, χωρίς αυτό να επηρεάζει εγγραφές που σχετίζονται με αυτή. Εδώ, η ιεράρχηση των εγγραφών μπορεί να επιτευχθεί με την ύπαρξη ενός μοναδικού αναγνωριστικού, το οποίο δεν απαιτείται να έχει νόημα εκτός των ορίων της βάσης. Αυτό φυσικά δεν είναι το ίδιο αποδοτικό όσο η έμφυτη δυνατότητα ιεράρχησης των δεδομένων που έχουν οι υπηρεσίες καταλόγου, αλλά μπορεί να φανεί ιδιαίτερα χρήσιμο όταν τα δεδομένα αλλάζουν συχνά. Εντούτοις, αν το σύστημα έχει αυξημένες απαιτήσεις μετονομασίας, μετακίνησης, και εκτέλεσης λειτουργιών που κάνουν χρήση αναδρομής στα δεδομένα, τότε η χρήση σχεσιακής βάσης δεδομένων ενδείκνυται. Αυτό φαίνεται φυσικό, καθώς οι σχεσιακές βάσεις είναι υλοποιημένες ώστε να μπορούν να χειρίζονται αποδοτικά αυτά τα είδη λειτουργιών. Τέλος, πολλές από τις υλοποιήσεις τους διαθέτουν βελτιστοποιημένες λειτουργίες συναλλαγών, οι οποίες μπορούν να εφαρμοστούν και σε κατανεμημένα περιβάλλοντα. Παρόλα αυτά, κάνουν χρήση πολύπλοκων μεθόδων πρόσβασης στα δεδομένα, όπως δομημένες γλώσσες επερωτήσεων (*SQL*). Αυτό ενδέχεται να επιβαρύνει την απόδοση της ομοσπονδίας, στην οποία οι αιτήσεις είναι επιθυμητό να έχουν όσο το δυνατόν χαμηλότερη πολυπλοκότητα.

Μελετώντας τις λειτουργίες που απαιτείται να επιτελεί το σύστημα που περιγράφουμε, και τις υπάρχουσες διαθέσιμες τεχνολογίες, διαπιστώθηκε πως η πλέον αποδοτική υλοποίηση με δυνατότητες κλιμάκωσης, θα αποτελείτο από ένα μηχανισμό χειρισμού αιτημάτων, την

εξυπηρέτηση των οποίων θα αναλάμβανε μια σχεσιακή βάση δεδομένων που θα υποστήριζε, μεταξύ άλλων, την ιεραρχία που τηρείται από μια υπηρεσία καταλόγου όπως το LDAP, το οποίο θα αποτελούσε ένα πολύ καλό παράδειγμα πρωτοκόλλου επικοινωνίας για το σύστημα [FedFS-NSDB]. Ωστόσο, αυτό δεν συνεπάγεται ότι μια σχεσιακή βάση είναι απαραίτητη για την υλοποίηση ενός ομοσπονδιακού συστήματος, όπως επίσης πως η μοναδική εναλλακτική βρίσκεται στις υπηρεσίες καταλόγου. Αντί αυτού, υποδεικνύει πως με τη χρήση του μοντέλου που περιγράψαμε, μπορούν να επιτευχθούν πιο αποδοτικές υλοποιήσεις του συστήματος.

2.5 Προσεγγίσεις Μηχανισμού Αντιστοίχισης

Κλείνοντας το κεφάλαιο, παραθέτουμε τρεις προσεγγίσεις της υπηρεσίας δεδομένων ονοματοχώρου (NSDB). Η αντιστοίχιση μπορεί να γίνει με οποιονδήποτε από τους τρόπους που θα περιγραφούν στη συνέχεια, και στην υλοποίηση χρησιμοποιείται ο τελευταίος. Σε αυτό το σημείο αξίζει να τονίσουμε έναν σημαντικό διαχωρισμό παραθέτοντας δύο ορισμούς:

- Η **υπηρεσία δεδομένων ονοματοχώρου**, ή υπηρεσία NSDB, αντιστοιχίζει κάθε FSN σε ένα ή περισσότερα FSL, ενώ ενδέχεται να εκτείνεται σε περισσότερους από έναν κόμβους.
- Ως **κόμβος NSDB** ορίζεται ένας τύπος εξυπηρετητή, ο οποίος υλοποιεί ένα τμήμα της υπηρεσίας δεδομένων ονοματοχώρου, και είναι υπεύθυνος να διαχειρίζεται τα FSL και τις σχετικές τους πληροφορίες που αντιστοιχούν σε ένα τμήμα των FSN της ομοσπονδίας.

2.5.1 Πρότυπο αρχείο εξαγωγής τοποθεσιών

Η πιο απλή υλοποίηση της υπηρεσίας αντιστοίχισης είναι η χρήση του κόμβου NSDB για τη σύνθεση ενός πρότυπου αρχείου τοποθεσιών για κάθε πελάτη, το οποίο περιέχει τις πραγματικές τοποθεσίες των συνόλων αρχείων που ο πελάτης χρειάζεται να προσαρτήσει ώστε να έχει πρόσβαση στον ομοσπονδιακό ονοματοχώρο.

Σε κάθε κόμβο NSDB, θεωρούμε ότι συγκρατούνται *πληροφορίες αντιστοίχισης* για όλα τα δεδομένα της ομοσπονδίας, οι οποίες συνδυάζουν τα μοναδικά αναγνωριστικά των συνόλων αρχείων με τις πραγματικές τους τοποθεσίες. Αν υποθέσουμε ότι υπάρχει κλιμάκωση στους κόμβους NSDB ώστε καθένας να διατηρεί αντιστοιχίσεις για ένα τμήμα του ομοσπονδιακού ονοματοχώρου, τότε πρέπει να εξασφαλιστεί ότι υπάρχει επαρκής πληροφορία σε όλους τους κόμβους, για την εύρεση εκείνου που διατηρεί την απαιτούμενη πληροφορία. Έτσι, η υπηρεσία που υλοποιείται στον κόμβο μπορεί να αναζητήσει όλες τις πληροφορίες για τα σύνολα αρχείων που αφορούν τον πελάτη, και να συνθέσει έπειτα ένα αρχείο κειμένου που θα τις περιέχει, το οποίο τελικά θα του μεταβιβάσει. Η λειτουργία αυτή μπορεί να εφαρμοστεί και για τους ομοσπονδιακούς εξυπηρετητές. Σε αυτή την περίπτωση, οι ριζικοί εξυπηρετητές μπορούν να βεβαιώνονται ότι διατηρούν το ριζικό σύνολο αρχείων με την σωστή μορφή, ενώ

οι υπόλοιποι μπορούν να ενημερώνονται για την εγκυρότητα των πληροφοριών που διαθέτουν στα σημεία προσάρτησης των συνόλων αρχείων που εξάγουν, διαφυλάσσοντας έτσι την συνοχή του ομοσπονδιακού ονοματοχώρου.

Αυτός ο τρόπος υλοποίησης επιτρέπει σε όλους τους κόμβους της ομοσπονδίας, ακόμα και στους πελάτες της, να έχουν δυναμικά αρχεία προσάρτησης του ονοματοχώρου, αν και αυτό δεν θα πρέπει να απαιτείται για τη σύνδεση στον ομοσπονδιακό ονοματοχώρο. Αν και η ορθότητα αυτής της προσέγγισης γίνεται εύκολα αντιληπτή, το ίδιο εύκολα γίνεται αντιληπτή και η μειωμένη απόδοση που αυτή συνεπάγεται. Μεταξύ του NSDB και του εξυπηρετητή γίνεται ανταλλαγή αρχείων, το μέγεθος καθώς και η μορφή των οποίων είναι απροσδιόριστα. Η προσέγγιση αυτή δεν εγγυάται καμία ευέλικτη μορφή ελέγχου στη σύνταξη των εγγραφών, χωρίς να εξετάζεται το θέμα της ασφάλειας. Τέλος, στο αρχείο πρέπει να επιτελούνται με ευκολία και αυξημένη αποδοτικότητα λειτουργίες αναζήτησης, προσθήκης και αφαίρεσης δεδομένων, λειτουργίες οι οποίες στην εν λόγω υλοποίηση δεν είναι αποδοτικά υλοποιήσιμες.

2.5.2 Χρήση Σχεσιακής Βάσης Δεδομένων

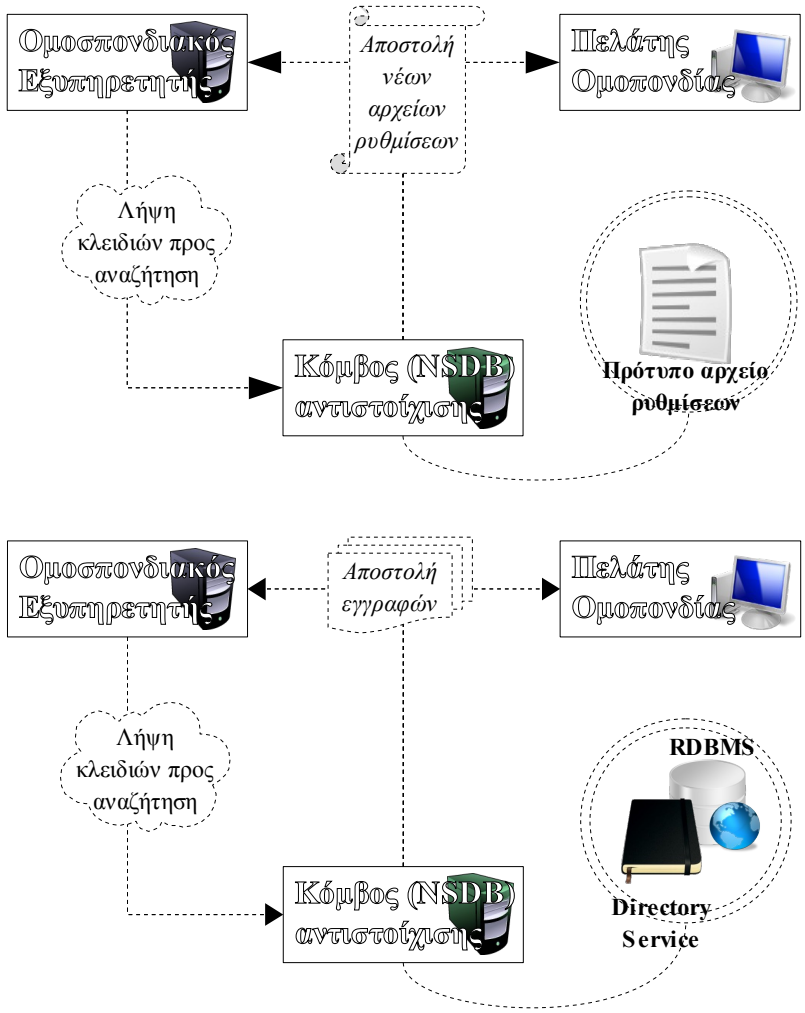
Η προσέγγιση που προηγήθηκε δεν μπορεί να θεωρηθεί κομψή επίλυση του προβλήματος της δημιουργίας ομοσπονδιακού ονοματοχώρου, καθώς προσφέρει χαμηλή αποδοτικότητα, και μεγάλο φόρτο επεξεργασίας και δεδομένων όχι μόνο στους εξυπηρετητές, αλλά και στο δίκτυο. Οι πόροι των ομοσπονδιακών κόμβων σπαταλώνται στην επεξεργασία αρχείων κειμένου, υποβαθμίζοντας πιθανώς τις δυνατότητες του συστήματος.

Εναλλακτικά, η αντιστοίχιση μεταξύ των ονομάτων και των τοποθεσιών ενός συνόλου αρχείων θα μπορούσε να επιτευχθεί με μια βάση δεδομένων. Η μοναδικότητα των κλειδιών που χαρακτηρίζουν τα ονόματα FSN έχει τώρα σημασία, καθώς επιτρέπει στην βάση να τα χρησιμοποιήσει στη συσχέτιση με τα κλειδιά των τοποθεσιών FSL, τα οποία θεωρούνται επίσης μοναδικά. Επιπλέον, οι λειτουργίες ευρετηρίου μπορούν να υλοποιήσουν ταχύτατα λειτουργίες αναζήτησης στην βάση, παρόλο που με τη χρήση ευρετηρίων, οι λειτουργίες προσθήκης και αφαίρεσης γίνονται χρονικά ακριβές λόγω των ενημερώσεων που προκύπτουν για τα ευρετήρια. Όμως η χρήση ευρετηρίων αποτελεί μονόδρομο, καθώς χωρίς αυτά η αναζήτηση μετατρέπεται σε ιδιαιτέρως χρονοβόρα λειτουργία. Αυτό δεν είναι αποδεκτό, καθώς τα δεδομένα της ομοσπονδίας θεωρούνται κατά προσέγγιση στατικά, εφόσον η λειτουργία που υπερισχύει στο σύστημα με διαφορά είναι η αναζήτηση.

2.5.3 Χρήση υπηρεσίας καταλόγου

Η ιδιότητα που διατυπώθηκε για τη στατική φύση των δεδομένων είναι αρκετή για να μας οδηγήσει να χρησιμοποιήσουμε μια υπηρεσία καταλόγου για την υλοποίηση των NSDB

κόμβων ενός ομοσπονδιακού συστήματος. Ιδιαίτερα σημαντικό θεωρούμε πως οι υπηρεσίες αυτές έχουν αυξημένη απόδοση σε λειτουργίες αναζήτησης. Η προσέγγιση είναι παρόμοια με την προηγούμενη, με την σημαντική διαφορά πως η αντιστοίχιση των δεδομένων στον κόμβο NSDB γίνεται από μια βάση όχι απαραίτητα σχεσιακού τύπου, την διαχείριση της οποίας αναλαμβάνει η διεπαφή της υπηρεσίας καταλόγου. Η προσέγγιση αυτή θεωρείται ιδιαίτερα δημοφιλής στην ερευνητική ομάδα του FedFS [FedFS-NSDB]. Η διεπαφή LDAP που προτείνεται από την ομάδα είναι η πλέον διαδεδομένη και μεταφέρσιμη υπηρεσία καταλόγου, και έτσι θεωρείται λογική η χρήση της. Η υπηρεσία καταλόγου LDAP χρησιμοποιείται και στην υλοποίηση που θα παρουσιαστεί στην συνέχεια.



Σχήμα 2.4: Σχηματική αναπαράσταση των στρατηγικών υλοποίησης του μηχανισμού αντιστοίχισης. Στην περίπτωση που ο κόμβος NSDB αναλαμβάνει την σύνταξη αρχείων ρυθμίσεων με βάση κάποιο πρότυπο, οι πληροφορίες που αποστέλλονται στους κόμβους τις ομοσπονδίας είναι αρχεία κειμένου, τα οποία εξάγονται από τον NSDB προς πελάτες και εξυπηρετητές. Τα αρχεία αυτά ανασυντάσσονται κατόπιν αίτησης κάποιου εξυπηρετητή, ο οποίος συμπεριλαμβάνει σε αυτή κλειδιά συνόλων αρχείων. Στην περίπτωση χρήσης κάποιας υπηρεσίας καταλόγου ή/και σχεσιακής βάσης δεδομένων οι κόμβοι ανταλλάσσουν πληροφορία με τη μορφή εγγραφών, οι οποίες ζητούνται από τους εξυπηρετητές της ομοσπονδίας.

3 *Αρχιτεκτονική Ομοσπονδίας*

"αρχιτεκτονική", (η) ουσ. *η τέχνη του αρχιτέκτονα*
|| εύρυθμη διάρθρωση λογοτεχνικού
ή καλλιτεχνικού έργου

Στο κεφάλαιο που προηγήθηκε έγινε εκτενής εισαγωγή στη σύνθεση και στις αρχές που διέπουν τα ομοσπονδιακά συστήματα. Κατά συνέπεια, στις ενότητες που ακολουθούν θα γίνει μια προσπάθεια μετάβασης σε πιο τεχνικά θέματα, ούτως ώστε να προσεγγιστεί ο σχεδιασμός πίσω από την υλοποίηση. Ακολουθεί μια περιγραφή υποθέσεων και ορισμών που περιορίζουν τον χώρο των λύσεων, για να γίνει περισσότερο κατανοητό το πρόβλημα και ο τρόπος υλοποίησής του.

3.1 Αρχιτεκτονικό Υπόβαθρο

Πριν διατυπωθούν οι στόχοι της υλοποίησης που ακολουθεί, επαναδιατυπώνουμε τους ορισμούς των συστατικών ενός ομοσπονδιακού συστήματος λαμβάνοντας υπόψιν τεχνικές λεπτομέρειες, ώστε να τεθούν απαιτήσεις και περιορισμοί που αφορούν την υλοποίηση που θα παρατεθεί.

Πρωτόκολλο εξαγωγής δεδομένων. Το πρωτόκολλο που χρησιμοποιείται για εξαγωγή δεδομένων στην ομοσπονδία δεν απαιτείται να είναι κοινό για όλους τους κόμβους της. Ανεξαρτήτως του αριθμού των πρωτοκόλλων που χρησιμοποιούνται όμως, θεωρούμε ότι κάθε ένα από αυτά υποστηρίζει στην υλοποίησή του, τα χαρακτηριστικά που διατυπώθηκαν αναλυτικά στον προηγούμενο κεφάλαιο. Για την υλοποίηση που ακολουθεί, λόγω ανυπαρξίας σύγχρονων πρωτοκόλλων που να ικανοποιούν τις απαιτήσεις αυτές, χρησιμοποιούμε το NFS

πρωτόκολλο, όπως αυτό ορίζεται στην τέταρτη έκδοσή του [Pawlowski00], καθώς οι στόχοι του θεωρούνται κοινοί σε πολλά σημεία με αυτούς ενός ομοσπονδιακού συστήματος. Επίσης, όπως θα γίνει εμφανές στη συνέχεια, έχει ήδη υλοποιημένα χαρακτηριστικά και δομές που βοηθούν σημαντικά στη δημιουργία ενός ενιαίου ονοματοχώρου.

Ομοσπονδιακοί εξυπηρετητές. Οι εξυπηρετητές της ομοσπονδίας, όπως έχουμε δει, χωρίζονται σε εξυπηρετητές αρχείων, δεδομένων ονοματοχώρου και ριζικούς εξυπηρετητές. Σε όλες τις περιπτώσεις έχουν τη δυνατότητα να αποθηκεύουν φυσικά δεδομένα συνόλων αρχείων ή πληροφοριών που τα αφορούν, καθώς και σημεία προσάρτησης προς άλλους ομότιμους εξυπηρετητές. Θεωρούμε ότι εξυπηρετούν όσα πρωτόκολλα εξαγωγής δεδομένων υποστηρίζονται από την ομοσπονδία, αν και στην υλοποίηση θα δούμε ότι γίνεται χρήση μόνο των NFSv4 και LDAP. Οι *εξυπηρετητές αρχείων* είναι αρμόδιοι μόνο για την εξαγωγή των αρχείων τους, και την ενημέρωση των σχετικών κόμβων της υπηρεσίας αντιστοίχισης για αυτά. Οι *ριζικοί εξυπηρετητές* της ομοσπονδίας, πέραν των απαιτήσεων που τέθηκαν για αυτούς σε προηγούμενες ενότητες, θεωρούμε ότι μπορούν να εξάγουν μόνο το ριζικό σύνολο αρχείων, ενώ ο τρόπος εντοπισμού και αναγνώρισης τους από τους πελάτες δεν καθορίζονται από το ομοσπονδιακό πρωτόκολλο, το οποίο υποθέτει ότι τουλάχιστον ένας από αυτούς είναι γνωστός εκ των προτέρων για κάθε πελάτη. Το *ριζικό σύνολο αρχείων* θεωρείται ότι είναι εικονικό, δηλαδή για λόγους απλότητας δεν περιέχει αρχεία, παρά μόνο μια δομή καταλόγων και σημείων προσάρτησης άλλων συνόλων αρχείων. Είναι μοναδικό, και αναπαράγεται σε κάθε ριζικό εξυπηρετητή μέσω του πρωτοκόλλου της ομοσπονδίας, ενώ αποτελεί το μοναδικό σημείο πρόσβασης στην ρίζα του ομοσπονδιακού ονοματοχώρου. Τέλος, οι *κόμβοι NSDB* απαρτίζουν την υπηρεσία δεδομένων ονοματοχώρου της ομοσπονδίας. Θεωρούμε ότι περιέχουν ένα τμήμα ή ολόκληρο το σύνολο αναγνωριστικών των συνόλων αρχείων (FSN) της ομοσπονδίας, αλλά όχι απαραίτητα και τις τοποθεσίες που αντιστοιχούν στο καθένα (FSL). Στην περίπτωση που η ζητούμενη τοποθεσία δεν περιέχεται στον κόμβο NSDB, τότε διατίθεται η πληροφορία του επόμενου ομότιμου κόμβου στον οποίο πρέπει να αναζητηθεί.

Αναγνωριστικά συνόλων αρχείων. Θεωρούμε ότι τα αναγνωριστικά ονόματα συνόλων αρχείων (FSN), είναι μοναδικά κατά μήκος ολόκληρης της ομοσπονδίας και συνοδεύονται πάντοτε από τη διεύθυνση του κόμβου NSDB, στον οποίο θα πρέπει να αναζητηθεί η τοποθεσία αποθήκευσής τους. Η ονομασία FSN δεν είναι περιοριστικό να αποθηκεύεται στον κόμβο NSDB που την συνοδεύει, όμως ο κόμβος αυτός πρέπει να μπορεί να αποτελέσει εφαλτήριο για την αναζήτηση και εύρεση του κόμβου που περιέχει την πληροφορία. Όσον αφορά τα αναγνωριστικά των τοποθεσιών, θεωρούμε πως και αυτά θα πρέπει να είναι μοναδικά ανά την ομοσπονδία. Παρόλα αυτά, η απαίτηση να διαφέρουν ένα προς ένα και από τα αναγνωριστικά που χρησιμοποιούνται για τον διαχωρισμό των συνόλων αρχείων είναι

προαιρετική, καθώς πιθανή ταύτιση δεν θα πρέπει να δημιουργεί πρόβλημα στην εύρυθμη λειτουργία του μηχανισμού. Επιπλέον, κάθε τοποθεσία οφείλει να συνοδεύεται και από χαρακτηριστικά πρόσβασης όπως η διεύθυνση, η διαδρομή και η θύρα του εξυπηρετητή από τον οποίο είναι διαθέσιμο το σύνολο αρχείων, το πρωτόκολλο που χρησιμοποιείται για την πρόσβαση στα δεδομένα, καθώς και κάποια προαιρετικά σχόλια που περιγράφουν το σύνολο αρχείων στην τρέχουσα τοποθεσία. Στην υλοποίηση που επιχειρείται, ο τύπος του πρωτοκόλλου παραλείπεται, καθώς χρησιμοποιείται μόνο το πρωτόκολλο NFSv4. Κάθε εξυπηρετητής πρέπει να μπορεί να βρει, δοθέντος ενός σημείου προσάρτησης, το FSN για το ζητούμενο σύνολο αρχείων και κατόπιν αίτησης στον τοπικό κόμβο NSDB, τουλάχιστον μια από τις τοποθεσίες FSL που αντιστοιχούν σε αυτό.

Ομοσπονδιακοί πελάτες. Οι πελάτες της ομοσπονδίας θεωρούμε ότι είναι απλοί πελάτες πρωτοκόλλων που υποστηρίζονται από την ομοσπονδία και έχουν πρόσβαση στα αρχεία δεδομένων διασχίζοντας τον ομοσπονδιακό ονοματοχώρο και επιτελώντας όλες τις γνωστές λειτουργίες στα αρχεία του, μεταβάλλοντας τα περιεχόμενα τους, δεδομένου ότι διαθέτουν τα απαραίτητα δικαιώματα. Στην περίπτωση της υλοποίησης που παρατίθεται, οι πελάτες της ομοσπονδίας κάνουν χρήση μόνο του πρωτοκόλλου NFSv4. Οι πελάτες του συστήματος θα πρέπει να έχουν τη δυνατότητα να επηρεάσουν τη λογική παρουσίαση των δεδομένων του ονοματοχώρου μέσω εκτέλεσης λειτουργιών σε τμήματά του, όπως δημιουργία, διαγραφή, μετονομασία, μετακίνηση κλπ. Η ομοσπονδία δεν έχει ευθύνη για τη διατήρηση συνέπειας στον ονοματοχώρο όπως τον αντιλαμβάνονται οι πελάτες, καθώς εφόσον διαφορετικές οντότητες μπορούν να διαχειρίζονται διαφορετικά μέρη του ονοματοχώρου, καθίσταται πιθανό ο πελάτης να έχει πρόσβαση σε τμήμα του, διαχειριζόμενο από κάποια οντότητα, το οποίο έχει κριθεί πεπαλαιωμένο στην περίπτωση που ένα άλλο ιεραρχικά ανώτερο τμήμα του, διαχειριζόμενο από κάποια άλλη οντότητα, έχει αλλάξει. Στην παρούσα φάση μπορούμε μόνο να θεωρήσουμε πως οι πελάτες ανακαλύπτουν τον ονοματοχώρο δυναμικά διασχίζοντας τον, χωρίς να τίθεται κάποια απαίτηση για τον έλεγχο των τμημάτων που έχουν ήδη παρέλθει της διάσχισης. Θα πρέπει, όμως, να δίνεται δυνατότητα στους πελάτες να διασχίζουν τα δεδομένα που εξάγονται από τους εξυπηρετητές έχοντας μονάχα γνώση της θέσης κάποιου ριζικού εξυπηρετητή, ώστε να προσαρτηθεί το ριζικό σύνολο από αυτόν. Δεν θα πρέπει να απαιτείται ειδική στατική ρύθμιση στους πελάτες, ενώ η διαδικασία εντοπισμού της τοποθεσίας των δεδομένων θα πρέπει να είναι διαφανής για αυτούς.

3.2 Στόχοι Υλοποίησης

Αδυναμίες σημερινών συστημάτων. Σήμερα, η δημιουργία ενός κοινού ονοματοχώρου καθίσταται δύσκολη για δύο σημαντικούς λόγους. Αφενός οι εξυπηρετητές που εξάγουν

αρχεία ενδέχεται να μην ανήκουν στον δικτυακό χώρο του διαχειριστή, και αφετέρου δεν υπάρχει κάποιος μηχανισμός, ο οποίος να δύναται να συνδυάσει τα εξαγόμενα δεδομένα, ώστε να προκύψει ένας ομοσπονδιακός ονοματοχώρος όπως ορίστηκε. Η στήριξη στον διαχειριστή του συστήματος ή σε ειδικά προσαρμοσμένο λογισμικό για τη σύνθεση του ονοματοχώρου, και η ανάγκη ενημέρωσης όλων των σχετικών αρχείων ρυθμίσεων των κόμβων του συστήματος σε περίπτωση μεταβολής της σύνθεσης αυτής, δεν αποτελούν σε καμία περίπτωση ανεκτές λύσεις. Επίσης υφίσταται το πρόβλημα χρήσης διαφορετικού λογισμικού μεταξύ των κόμβων, καθώς δεν υπάρχει πρωτόκολλο που να μπορεί να χρησιμοποιηθεί για την παροχή των υπηρεσιών του ονοματοχώρου σε κάθε σύστημα.

Στόχος. Ο στόχος της εργασίας είναι να επιλύσει τα προβλήματα αυτά. Η υλοποίηση που παρατίθεται αναλαμβάνει να δημιουργήσει έναν ομοσπονδιακό ονοματοχώρο που προκύπτει από πολλαπλούς εξυπηρετητές, και είναι κοινός για κάθε πελάτη του. Το σημαντικότερο χαρακτηριστικό του νέου συστήματος είναι πως επιτρέπει την εύκολη δημιουργία και διαχείριση του ονοματοχώρου, χωρίς να παραβιάζεται η ενιαία μορφή που τον διέπει. Στην υλοποίηση, όπως και σε παρόμοιες θεωρητικές περιγραφές [FedFS-NSDB],[Glamour05], χρησιμοποιούμε αποκλειστικά το NFSv4, καθώς καλύπτει τις απαιτήσεις της ομοσπονδίας. Για τη σύνθεση του ονοματοχώρου χρησιμοποιείται ο μηχανισμός αντιστοίχισης που υλοποιείται με χρήση της υπηρεσίας καταλόγου LDAP, η οποία προτιμάται λόγω της ευελιξίας που παρέχει στον χειρισμό της μορφής των αποθηκευόμενων δεδομένων και της μεταφερσιμότητας της, που αποτελεί σημαντικό χαρακτηριστικό για την ομοσπονδία. Τέλος, για την ολοκλήρωση του ονοματοχώρου χρησιμοποιείται μηχανισμός, ο οποίος αναλαμβάνει να ενημερώνει τους ριζικούς εξυπηρετητές για τη μορφή του ριζικού συνόλου αρχείων, ώστε να είναι κοινή για όλους. Παρόλα αυτά, λόγω ελλείψεων στο θεωρητικό υπόβαθρο αναγκαζόμαστε να θεωρήσουμε ότι οι πληροφορίες του ριζικού συνόλου αποθηκεύονται σε ενδεδειγμένους κόμβους αντιστοίχισης, και έτσι στην υλοποίηση έχουμε χρησιμοποιήσει ενδεικτικά έναν κόμβο ο οποίος ακολουθεί αυτή την υπόθεση. Τελικά, οι διαχειριστικές εργασίες πάνω στον ονοματοχώρο του συστήματος που δημιουργήθηκε είναι ιδιαίτερα προσιτές, καθώς αντί χρήσης μηχανισμών ενημέρωσης των αρχείων ρύθμισης κάθε κόμβου του συστήματος, ο ονοματοχώρος καταγράφεται σε βάσεις δεδομένων, με αποτέλεσμα αλλαγές στη μορφή του να απαιτούν μόνο ενημέρωση της πληροφορίας σε αυτές.

3.3 Αρχιτεκτονικός Σχεδιασμός

3.3.1 Χρησιμοποιούμενες τεχνολογίες

Όπως αναφέρθηκε και στην εισαγωγή, η ανάπτυξη τεχνολογιών για τη δημιουργία κατανεμημένων συστημάτων δεν είναι πρόσφατο επίτευγμα της πληροφορικής. Η διάρκεια

που έχουν σημειώσει πολλές από αυτές, τις καθιστά αρκετά ασφαλείς θεμέλιους λίθους για τη δημιουργία ενός νέου συστήματος, καθώς έχουν χρησιμοποιηθεί με επιτυχία σε ετερογενή παραγωγικά περιβάλλοντα. Σημαντικός λόγος εξέτασής αυτών που θα παρουσιαστούν, είναι ότι καλύπτουν τις ανάγκες που ανακύπτουν κατά τη δημιουργία ενός ομοσπονδιακού συστήματος, με αποτέλεσμα να γίνεται χρήση τους σε θεωρητικές, αλλά και πρακτικές προσπάθειες υλοποίησης ομοσπονδιακών συστημάτων. Αυτό συμβαίνει λόγω του γεγονότος πως ικανοποιούν εξ' ορισμού ένα κομμάτι των απαιτήσεων που θέτει ένα τέτοιο σύστημα, αλλά και επειδή οι στόχοι ανάπτυξης που θέτουν, συμβαδίζουν με αυτούς των ομοσπονδιακών συστημάτων.

Network Filesystem. Το πρωτόκολλο NFS θεωρείται θεμέλιος λίθος των κατανεμημένων συστημάτων. Όμως, ο σχεδιασμός του είναι περιορισμένος για τα ομοσπονδιακά συστήματα, καθώς έχει σχεδιαστεί για χρήση σε τοπικά δίκτυα. Η ασφάλεια που το χαρακτηρίζει είναι προσανατολισμένη σε μικρές ομάδες εργασίας, και έτσι έχει σημαντικά κενά, τα οποία θα ήταν ανεπίτρεπτα σε μεγαλύτερη κλίμακα. Στις περισσότερες υλοποιήσεις η υπηρεσία αναπαραγωγής είναι ανύπαρκτη, ενώ δεν υποστηρίζονται οι αναφορικές υπερσυνδέσεις. Παρά ταύτα η τέταρτη έκδοσή του, το NFSv4 [Pawlowski00], εστιάζει στα προβλήματα που δημιουργούνται σε απόδοση και ασφάλεια από τη χρήση προηγούμενων εκδόσεών του σε δίκτυα ευρείας περιοχής και τη λειτουργία σε διαφορετικούς δικτυακούς χώρους (*domains*). Παρέχει λύση στο πρόβλημα δημιουργίας κοινού, ενιαίου ονοματοχώρου υποστηρίζοντας ένα εικονικό σύστημα αρχείων, το οποίο παρόμοια με το ριζικό σύνολο αρχείων, αναλαμβάνει να αποτελέσει τη ρίζα του κοινού ονοματοχώρου. Τα δεδομένα εξάγονται ιεραρχικά από κάθε εξυπηρετητή, κάτι που επιτρέπει την ασύνδετη εξαγωγή δεδομένων από έναν οργανισμό, αλλά ταυτόχρονα διευκολύνει τη σύνθεση του ονοματοχώρου. Γενικότερα στηρίζεται στους ίδιους μηχανισμούς χειρισμού ονομασιών τοποθεσιών που χρησιμοποιούν και οι πρόγονοί του. Σημαντικό είναι το γεγονός πως προσφέρει την δυνατότητα στον εξυπηρετητή να χρησιμοποιήσει αναφορικές υπερσυνδέσεις. Έτσι, δίνεται η δυνατότητα να συνδεθούν διαφορετικά συστήματα αρχείων, αλλά παρατηρείται έλλειψη σε πρότυπα ικανά να καλύψουν τις απαιτήσεις που θέτει ένας ονοματοχώρος ως προς την πληροφορία που πρέπει να μεταφερθεί. Το NFSv4 έχει πολλούς κοινούς στόχους με αυτούς που τέθηκαν στην εισαγωγή, και αυτό το καθιστά ικανή επιλογή για την υλοποίηση που επιχειρούμε.

Lightweight Directory Access Protocol. Το LDAP αποτελεί την πλέον υποσχόμενη υπηρεσία καταλόγου, επομένως είναι αναμενόμενο να εξετάζεται η πιθανότητα χρήσης του για την υλοποίηση ενός ομοσπονδιακού συστήματος. Με τη χρήση των υπαρχόντων εγκαταστάσεων του LDAP σε οργανισμούς και επιχειρήσεις μπορούμε να μειώσουμε το κόστος συντήρησης, καθώς και να βοηθήσουμε να παγιωθεί η υποδομή πληροφοριών και

τεχνολογίας του συστήματος. Το LDAP καθορίζει ένα πρωτόκολλο επικοινωνίας για την πρόσβαση και την αναζήτηση σε μια βάση εγγραφών που αποτελούνται από γνωρίσματα. Δεν καθορίζει τόσο την υπηρεσία καταλόγου, όσο τη μορφή και τη μεταφορά μηνυμάτων, μέσω των οποίων επιτυγχάνεται η επικοινωνία των πελατών με τη βάση. Τα μηνύματα αυτά σχετίζονται με τις εγγραφές της βάσης, των οποίων η δομή, αλλά και οι σχέσεις συσχέτισης καθορίζονται από σχετικά ευέλικτα σχήματα (*schemas*). Δυστυχώς οι υλοποιήσεις του LDAP που υφίστανται μέχρι στιγμής, δεν επιτρέπουν τη μετονομασία των εγγραφών της βάσης, τουλάχιστον σε επίπεδο που μπορεί να θεωρηθεί αποδεκτό αποδοτικά. Λόγω της εξάρτησης κάθε εγγραφής από μια αναγνωριστική ονομασία, η οποία μεταξύ άλλων καθορίζει και τη θέση της στην ιεραρχία, οποιαδήποτε τροποποίηση θα οδηγούσε και σε αλλαγή της θέσης αυτής στο σύστημα. Ωστόσο, το πρόβλημα αυτό δεν φαίνεται να είναι απαγορευτικό για τον χρήστη της υπηρεσίας, καθώς μπορούμε να θεωρήσουμε ότι σε μεγάλο βαθμό, τα δεδομένα που αποθηκεύονται στη βάση και αφορούν τις θέσεις των δεδομένων στην ομοσπονδία, μπορούν να ιδωθούν ως στατικά.

3.3.2 Αρχιτεκτονικός Σχεδιασμός Υλοποίησης

Το σύστημα που έχουμε υλοποιήσει αποτελείται από τέσσερα είδη κόμβων: τους πελάτες, τους εξυπηρετητές αρχείων, τους ριζικούς εξυπηρετητές και τους κόμβους υπηρεσίας ονοματοχώρου. Σε όλους τους κόμβους του συστήματος εκτελείται λειτουργικό σύστημα Debian Linux, ο πυρήνας του οποίου έχει παραμετροποιηθεί ώστε να περιέχει υποστήριξη για το πρωτόκολλο NFSv4, ενώ για τις ανάγκες τις υλοποίησης έχει γίνει παρέμβαση στον κώδικά του, ώστε να υποστηριχθεί ο μηχανισμός αναζήτησης. Περισσότερες λεπτομέρειες για την παρέμβαση στον κώδικα του πυρήνα δίνονται στο επόμενο κεφάλαιο.

Σχεδιασμός μηχανισμού αντιστοίχισης. Ο μηχανισμός αντιστοίχισης λειτουργεί όπως ορίστηκε στις απαιτήσεις. Το εκάστοτε αρχείο διαδρομών εξαγωγής (*export table*) διαθέτει τις τοποθεσίες που πρέπει να αναζητηθούν σε απομακρυσμένους κόμβους, συνοδευόμενες από τα FSN των συνόλων αρχείων στα οποία οδηγούν, και την διεύθυνση του κόμβου NSDB από τον οποίο μπορεί να ξεκινήσει η αναζήτηση της τοποθεσίας. Η αίτηση αναζήτησης γίνεται από τον πυρήνα του λειτουργικού συστήματος και απευθύνεται σε δαίμονα, ο οποίος βρίσκεται στο επίπεδο χρήστη του εκάστοτε κόμβου της υπηρεσίας αναζήτησης. Η επικοινωνία μεταξύ πυρήνα και δαίμονα γίνεται με την χρήση Unix sockets. Ο δαίμονας έχει το ρόλο του εξυπηρετητή, και ο πυρήνας ανοίγει μια σύνδεση για κάθε αίτηση αντιστοίχισης. Στην συνέχεια, ο δαίμονας αναλαμβάνει να επικοινωνήσει με την τοπική βάση LDAP, και να ανακτήσει τις ζητούμενες τοποθεσίες. Αν η αναζήτηση αποτύχει τοπικά, τότε ξεκινά η αναζήτηση της πληροφορίας με τη σύνδεση του δαίμονα και σε απομακρυσμένες βάσεις.

Όταν οι ζητούμενες πληροφορίες αντιστοίχισης ανακτηθούν, ο δαίμονας χρησιμοποιεί την σύνδεση που έχει επιτευχθεί με τον πυρήνα, για να του επιστρέψει τα αποτελέσματα. Από την στιγμή που οι πληροφορίες έχουν ληφθεί, ενσωματώνονται στις αντίστοιχες δομές του πρωτοκόλλου, και η ανάκτηση των ζητούμενων δεδομένων γίνεται με τα νέα στοιχεία που έχουν αποθηκευτεί εκεί. Λόγω του γεγονότος πως η υλοποίηση του μηχανισμού περιορίζεται στην παρέμβαση σε ένα μικρό τμήμα του πυρήνα που υλοποιεί το πρωτόκολλο NFSv4, και στη δημιουργία ενός δαίμονα σε επίπεδο χρήστη, το αποτέλεσμα είναι μεταφέρσιμο σε οποιαδήποτε διανομή linux υποστηρίζει πυρήνες που υλοποιούν τις λειτουργίες του NFSv4. Οι πυρήνες που θεωρείται ότι διαθέτουν τέτοια υποστήριξη ανήκουν αποκλειστικά στην οικογένεια 2.6, και ειδικότερα έχουν μεγαλύτερο αριθμό έκδοσης από 2.6.25, για τον οποίο διατίθενται ειδικά patches από την ερευνητική ομάδα στο πανεπιστήμιο του Michigan [CitiNFSProject]. Παρόλα αυτά, η υλοποίηση έχει δοκιμαστεί και στον πυρήνα 2.6.24 χωρίς επιπρόσθετες αλλαγές, όπου δεν παρατηρήθηκε πρόβλημα στην λειτουργία του συστήματος. Από αυτό συμπεραίνουμε πως η υποστήριξη όλων των χαρακτηριστικών του πρωτοκόλλου δεν είναι απαραίτητη για τη λειτουργία της υλοποίησης που παρουσιάζεται εδώ, αν και συνίσταται η νεότερη δυνατή έκδοση, λόγω των βελτιώσεων που παρέχει, μεταξύ άλλων, και στην ασφάλεια του συστήματος.

Σύνθεση ονοματοχώρου. Η σύνθεση των επιμέρους τμημάτων του ονοματοχώρου γίνεται από τις υπηρεσίες που περιγράψαμε και εξαρτάται σε μεγάλο βαθμό από την δυνατότητα να οριστεί η ρίζα του. Στην υλοποίηση που παρατίθεται, το ριζικό σύνολο αποτελείται μόνο από σημεία προσάρτησης που παραπέμπουν σε άλλα σύνολα αρχείων του ονοματοχώρου. Παρόλο που υφίσταται η δυνατότητα εξαγωγής ενός ολοκληρωμένου συνόλου αρχείων από τον ριζικό εξυπηρετητή, η παράλειψη στατικών αρχείων και φακέλων διευκολύνει ιδιαίτερα την αναπαραγωγή του συνόλου στους ριζικούς εξυπηρετητές. Οι πληροφορίες που αφορούν τη σύσταση του ριζικού συνόλου αποθηκεύονται στις βάσεις που περιέχουν και τα δεδομένα αντιστοίχισης του ονοματοχώρου. Στην παρούσα φάση, δεν έχουν προταθεί τρόποι επιλογής των κόμβων της υπηρεσίας ονοματοχώρου όπου θα αποθηκεύονται αυτά τα δεδομένα, και έτσι στην υλοποίηση επιλέγουμε τυχαία έναν κόμβο υπηρεσίας δεδομένων της ομοσπονδίας για αυτή τη χρήση. Αυτή η επιλογή δεν θεωρείται λύση στο πρόβλημα που υφίσταται, καθώς ένας από τους σκοπούς της ομοσπονδίας είναι η κατανομή διαχειριστικών δικαιωμάτων στους οργανισμούς, μόνο πάνω στα δεδομένα που έχουν εξάγει. Η κατοχή των κόμβων που περιέχουν τις πληροφορίες του ριζικού συνόλου από ορισμένους οργανισμούς, όμως, αντιλαμβανόμαστε ότι δημιουργεί σημαντικό προβάδισμα στη διαχείρισή του ονοματοχώρου, έναντι των υπόλοιπων οργανισμών. Τέλος, στους ριζικούς εξυπηρετητές υλοποιείται ο δαίμονας ριζικού συνόλου, ο οποίος εξετάζει περιοδικά τις πληροφορίες που διατίθενται για

το ριζικό σύνολο στις βάσεις και συνθέτει τα κατάλληλα αρχεία ρυθμίσεων για τον κόμβο στον οποίο εκτελείται, ούτως ώστε να εξάγεται πάντοτε η πλέον ενημερωμένη μορφή του ριζικού συνόλου αρχείων. Οι πελάτες της ομοσπονδίας αρκεί να προσαρτήσουν τη ρίζα του συνόλου των εξαγόμενων δεδομένων των εξυπηρετητών ώστε να έχουν πρόσβαση στον ονοματοχώρο. Σε αυτό το σημείο οφείλουμε να τονίσουμε ένα ακόμα σημείο αδυναμίας της υλοποίησης. Αν το ριζικό σύνολο αλλάξει μορφή, τότε μέσω του δαίμονα που περιγράψαμε, οι ριζικοί εξυπηρετητές θα το αντιληφθούν και θα ενημερώσουν τα αρχεία ρυθμίσεων και τα σημεία προσάρτησης τους. Όμως δεν υπάρχει κάποιος τρόπος να ενημερώσουμε και τους πελάτες. Μια προφανής λύση θα ήταν να αναγκάζουμε κάθε πελάτη να εξετάζει την μορφή του ονοματοχώρου στην οποία έχει πρόσβαση, και σε περίπτωση που εντοπιστεί ασυνέπεια με τον τρέχον ονοματοχώρο, να γίνεται επαναπροσάρτηση του ονοματοχώρου. Όμως αυτό μπορεί να δημιουργήσει σοβαρά προβλήματα στην περίπτωση που συμπέσει με εργασίες που εκτελούν οι πελάτες, σε ένα ορισμένο σημείο του ονοματοχώρου. Έτσι, επειδή ο τρόπος με τον οποίο μπορούμε να διακόψουμε την εργασία των πελατών για να τους ενημερώσουμε για τις αλλαγές που έχουν συντελεστεί στον ονοματοχώρο δεν είναι προφανής, αφήνουμε στην αρμοδιότητα των χρηστών την επαναπροσάρτηση του ονοματοχώρου. Λόγω του γεγονότος πως η αλλαγή του ονοματοχώρου δεν επηρεάζει άμεσα την εργασία ενός πελάτη σε κάποιο σύνολο αρχείων, αφού αποτελεί αυτόνομη οντότητα στον εξυπηρετητή όπου αποθηκεύεται, θα ήταν αρκετό για την επίλυση του προβλήματος να γίνεται έλεγχος χρήσης του κοινού ονοματοχώρου από τοπικούς χρήστες, και σε αδρανή διαστήματα να γίνεται η απαραίτητη ενημέρωση.

Υλοποίηση ομοσπονδίας. Η αρχιτεκτονική των κόμβων που χρησιμοποιούμε για την υλοποίηση φαίνεται στο Σχήμα 3.1. Οι κόμβοι δημιουργούνται ως εικονικά λειτουργικά συστήματα με τη βοήθεια του User Mode Linux [Dike06] που περιέχεται στον πυρήνα που χρησιμοποιούμε για τα πειράματα. Για την δοκιμή της ομοσπονδίας, χρησιμοποιήσαμε οκτώ κόμβους. Οι δύο εξ' αυτών αποτέλεσαν πελάτες στατικής ρύθμισης που προσαρτούσαν το ριζικό σύνολο αρχείων της ομοσπονδίας από το ριζικό εξυπηρετητή, ο οποίος υλοποιείτο σε άλλο κόμβο της ομοσπονδίας. Επίσης χρησιμοποιήσαμε τρεις εξυπηρετητές, εκ των οποίων οι δύο αναλάμβαναν να εξάγουν δύο διαφορετικά σύνολα αρχείων, τα οποία συνέθεταν τον ονοματοχώρο, ενώ ο τρίτος εξυπηρετητής αποτελούσε αντίγραφο του ενός, στο οποίο έπρεπε να απευθυνθεί ο πελάτης, σε περίπτωση που υπήρχε πρόβλημα στην επικοινωνία του με τον πρώτο εξυπηρετητή. Τα δεδομένα του ριζικού συνόλου του ονοματοχώρου αποθηκεύονταν σε διαφορετικό κόμβο, ο οποίος ήταν και ο κύριος κόμβος NSDB στον οποίο απευθύνονταν όλες οι αιτήσεις. Επιλέξαμε αυτή την προσέγγιση, ούτως να καλύψουμε όλα τα πιθανά σενάρια που μπορούν να προκύψουν στην ομοσπονδία, από την παραπομπή σε άλλο κόμβο

NSDB κατά την αναζήτηση τοποθεσίας συνόλου αρχείων, μέχρι την παραπομπή του πελάτη σε εξυπηρετητή που περιέχει αντίγραφο των δεδομένων, σε περίπτωση προβλήματος με τον πρώτο. Τα σενάρια αυτά δεν συμπεριλήφθηκαν στα πειράματα, όπου εξετάζεται η απόδοση του σε κανονικές συνθήκες λειτουργίας.

Όπως γίνεται αντιληπτό από την περιγραφή, ο πραγματικός διαχωρισμός των κόμβων γίνεται με βάση τις υπηρεσίες που υλοποιούνται σε αυτούς. Αυτό σημαίνει πως εφόσον οι υπηρεσίες που υλοποιούνται σε διαφορετικούς κόμβους μπορούν να συνδυαστούν, τότε ένας κόμβος μπορεί να έχει περισσότερους από έναν ρόλους στην ομοσπονδία. Παρόλο που κάτι τέτοιο δεν αναφέρεται στην εκφώνηση, στην υλοποίηση διαπιστώθηκε πως είναι δυνατόν ο ίδιος κόμβος να επιτελεί τους ρόλους του πελάτη, του εξυπηρετητή αρχείων και κόμβου της υπηρεσίας ονοματοχώρου ταυτόχρονα, χωρίς να δημιουργείται πρόβλημα, αν και το σενάριο αυτό δεν θεωρείται περισσότερο χρήσιμο, ή έστω πιθανό, σε πραγματικές συνθήκες.

* Στην περίπτωση που η αντιστοίχιση λαμβάνει χώρα στον πελάτη
** Στην περίπτωση που η αντιστοίχιση λαμβάνει χώρα στον εξυπηρετητή

Σχήμα 3.1: Σχηματική αναπαράσταση αρχιτεκτονικής ομοσπονδιακών κόμβων. Οι κόμβοι χωρίζονται σε τρία επίπεδα: το επίπεδο του πυρήνα, του λειτουργικού συστήματος και του δαίμονα αντιστοίχισης. Ένας πελάτης διαθέτει μόνο τα δύο πρώτα, και απαιτείται να χρησιμοποιεί παραμετροποιημένο πυρήνα μόνο σε περίπτωση που η αντιστοίχιση γίνεται σε αυτόν. Ο NSDB κόμβος απαιτείται να διαθέτει μόνο τον δαίμονα NSDBd και μια βάση LDAP με τις απαραίτητες πληροφορίες. Τέλος, οι εξυπηρετητές απαιτείται να διαθέτουν πυρήνα με παραμετροποιήσεις μόνο σε περίπτωση που αναλαμβάνουν την αντιστοίχιση. Διαφορετικά, το μόνο λογισμικό που απαιτείται είναι ο δαίμονας RFSd που θα πρέπει να βρίσκεται εγκατεστημένος στον ριζικό εξυπηρετητή.

4 *Υλοποίηση Ονοματοχώρου*

"υλοποίηση", (η) ουσ. *η μεταβολή σε ύλη*
|| (μτφ.) *πραγματοποίηση*

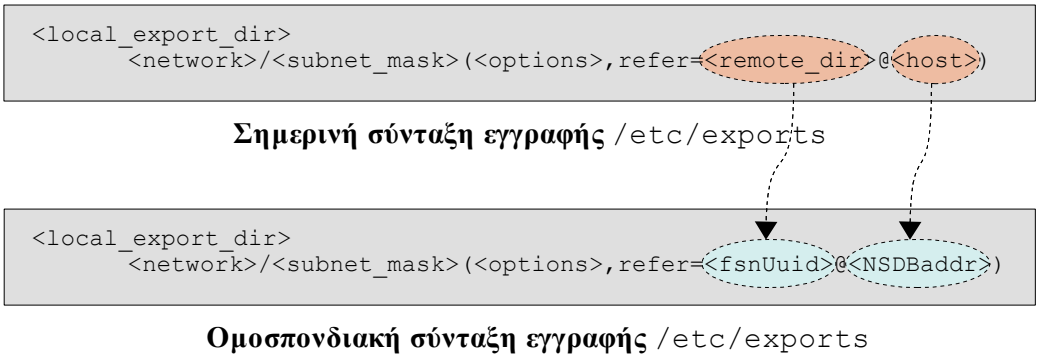
Έχουμε αναφερθεί ήδη εκτενώς στην δομή των ομοσπονδιακών συστημάτων, αλλά και στην αρχιτεκτονική της υλοποίησης. Παρόλα αυτά, οι τεχνικές λεπτομέρειες που επεξηγούν την κατασκευή των μερών της υλοποίησης έχουν παραλειφθεί στα προηγούμενα κεφάλαια, ώστε να αναλυθούν στο σύνολο τους εδώ.

4.1 Παραμετροποίηση Πυρήνα

Κωδικοποίηση XDR. Στο προηγούμενο κεφάλαιο έγινε αναφορά σε παραμετροποίηση που απαιτείται να γίνει στον πυρήνα του linux, ώστε οι λειτουργίες του πρωτοκόλλου NFSv4 να συμβουλεύονται τον εκάστοτε κόμβο της υπηρεσίας δεδομένων για να αντιστοιχίσουν το FSN του συνόλου αρχείων που διαθέτουν, με τις τοποθεσίες στις οποίες αυτό αποθηκεύεται. Η πληροφορία που μεταδίδεται μέσω του NFSv4 κωδικοποιείται στην πρότυπη μορφή XDR [RFC4506], ή οποία αναλαμβάνει να την μορφοποιήσει με προκαθορισμένο τρόπο, ούτως ώστε να εξασφαλιστεί η ανεξαρτησία της από το υλικό και το λειτουργικό σύστημα στο οποίο εκτελείται το NFSv4. Η αποκωδικοποίηση της πληροφορίας γίνεται στο άλλο άκρο της επικοινωνίας, στον παραλήπτη, όπου είτε αποθηκεύεται στις κατάλληλες δομές, είτε χρησιμοποιείται κατά απαίτηση.

Χρήση NFSv4. Για να γίνει κατανοητό το σκεπτικό πίσω από την υλοποίηση, πρέπει να εξηγήσουμε τον τρόπο με τον οποίο χρησιμοποιείται το πρωτόκολλο για τη μεταφορά και την

προσπέλαση των ζητούμενων δεδομένων. Όσο ο πελάτης βρίσκεται σε ένα σύνολο αρχείων, μπορεί να αιτηθεί την αποστολή και τη λήψη δεδομένων μέσω του πρωτοκόλλου, χωρίς αυτό να επηρεάζει τη λειτουργία της ομοσπονδίας. Όμως κατά τη διάσχιση ενός σημείου προσάρτησης προς ένα θυγατρικό ή πατρικό σύνολο[1] το πρωτόκολλο συμβουλεύεται τις αποθηκευμένες δομές τοποθεσιών και ανακτά το FSN του ζητούμενου συνόλου αρχείων. Αξίζει να σημειωθεί ότι οι δομές αυτές ενημερώνονται από αρχεία ρυθμίσεων[2], η ανάγνωση των οποίων επιτελείται κατά την κλήση υπεύθυνων υπηρεσιών, ώστε να ανακτηθούν οι πληροφορίες που περιέχουν. Μια από αυτές τις δομές είναι και η δομή *fs_locations*, όπως περιγράφεται θεωρητικά στο [RFC3530], στην οποία συγκρατούνται οι τοποθεσίες όπου μπορούν να βρεθούν τα δεδομένα που δεν ανήκουν σε σύνολο αρχείων του τρέχοντα εξυπηρετητή. Τα δεδομένα της δομής αυτής αντλούνται από το αρχείο /etc/exports, το οποίο συντάσσεται όπως φαίνεται στο Σχήμα 4.1. Όπως φαίνεται από το σχήμα, το αρχείο στην προκαθορισμένη μορφή του περιέχει τις διευθύνσεις IP των απομακρυσμένων εξυπηρετητών στις οποίες μπορούν να βρεθούν τα ζητούμενα σύνολα αρχείων, καθώς και τις σχετικές τοποθεσίες στις οποίες αποθηκεύονται εκεί, με σημείο αναφοράς τη ρίζα του εξαγόμενου ψευδο-συστήματος αρχείων. Στην υλοποίηση παραλλάσσουμε το αρχείο ώστε να περιέχει δύο διαφορετικές πληροφορίες που είναι: το FSN του ζητούμενου συνόλου στη θέση του μονοπατιού εξαγωγής, και η διεύθυνση του κόμβου υπηρεσίας ονοματοχώρου όπου μπορεί να αναζητηθεί η τοποθεσία του, η οποία αντικαθιστά τη διεύθυνση του εξυπηρετητή.



Σχήμα 4.1: Αναπαράσταση της μεταβολής που εφαρμόστηκε στη σύνταξη των εγγραφών του αρχείου /etc/exports, ώστε να χρησιμοποιηθεί στη δημιουργία ομοσπονδίας. Η διαφορά έγκειται στη σύνταξη της παραμέτρου refer που υλοποιεί την λειτουργία της αναφορικής υπερσύνδεσης. Συγκεκριμένα, αντί να λαμβάνονται ως ορίσματα ο εξυπηρετητής που φιλοξενεί τα δεδομένα και το σχετικό μονοπάτι σε αυτόν, τα νέα ορίσματα είναι το αναγνωριστικό του συνόλου αρχείων που περιέχει τα δεδομένα, και η διεύθυνση του NSDB κόμβου από όπου μπορεί να ξεκινήσει η αναζήτησή του. *Να σημειωθεί ότι κάθε πεδίο host μπορεί να περιέχει περισσότερες από μια διευθύνσεις, ενώ είναι επίσης δυνατό να έχουμε επαναλαμβανόμενο μοτίβο* "`refer=<remote_dir>@<host>;...;<remote_dir>@<host>`", *σε περίπτωση που τα υπόλοιπα αντίγραφα δεν εξάγονται από το ίδιο μονοπάτι στους αντίστοιχους εξυπηρετητές. Αυτή η ιδιότητα κληρονομείται και στην αλλαγή που έγινε, δίνοντας τη δυνατότητα να εισάγουμε περισσότερους από έναν NSDB, ή ακόμα και αντίγραφα του συνόλου αρχείων με διαφορετικό FSN, κάτι που δεν θεωρείται, όμως, χρήσιμο σε πραγματικές συνθήκες.*

[1]Η μετάβαση στο πατρικό σύνολο εκτελείται μέσω των συνδέσμων '.' και '..' στο linux.

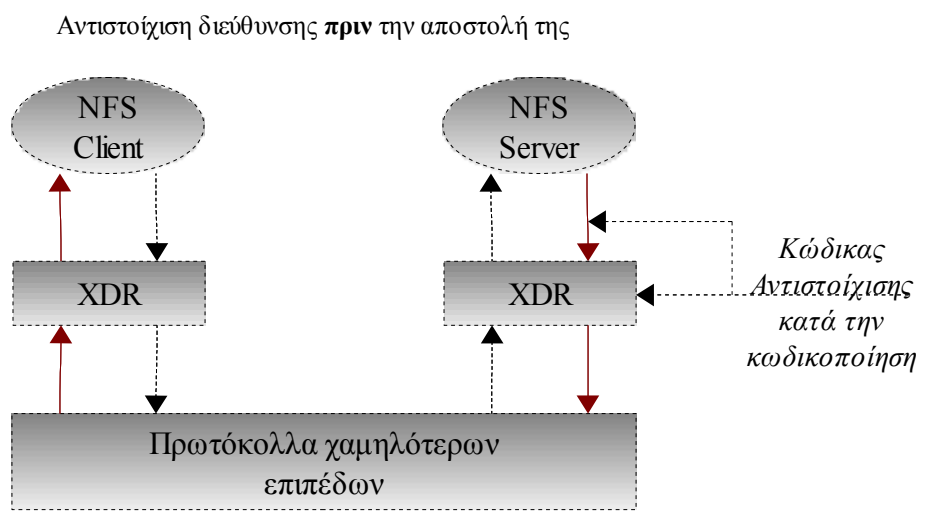
[2]Στο linux, τα αρχεία αυτά είναι αποτελούνται από το αρχείο /etc/exports που περιέχει τις διαδρομές των δεδομένων που εξάγονται, καθώς και τα αρχεία που αφορούν την ασφάλεια του πρωτοκόλλου, δηλαδή τα /etc/hosts.allow και /etc/hosts.deny

Τρόποι αντιστοίχισης. Το FSN ανακτάται αρχικά στον εκάστοτε εξυπηρετητή από το τοπικό αρχείο ρυθμίσεων που διαθέτει. Μόλις το αρχείο αναγνωσθεί, οι πληροφορίες του αποθηκεύονται στη δομή *fs_locations* του εξυπηρετητή. Σε εκείνη τη φάση, το πρωτόκολλο είναι έτοιμο να στείλει την πληροφορία στον πελάτη, όπου γίνεται η χρήση της. Η παραπάνω διαδικασία κάνει φανερό ότι υπάρχει η δυνατότητα είτε να γίνει αντιστοίχιση των πληροφοριών πριν αποσταλούν από τον εξυπηρετητή, είτε να αποσταλούν το FSN και η διεύθυνση του κόμβου NSDB στον πελάτη, και να αναλάβει αυτός την αντιστοίχιση τους. Οι δύο αυτοί τρόποι περιγράφονται στις επόμενες υποενότητες.

4.1.1 Αντιστοίχιση στον Εξυπηρετητή

Κατά την αποστολή της τοποθεσίας ενός συνόλου αρχείων, ο εξυπηρετητής παίρνει στο πρωτόκολλο τον ρόλο του αποστολέα και ο πελάτης τον ρόλο του παραλήπτη. Όπως φαίνεται στο Σχήμα 4.2, μια πιθανή επίλυση του προβλήματος είναι η παραμετροποίηση του NFSv4 ώστε ο εξυπηρετητής να αναλαμβάνει την αντιστοίχιση της ονομασίας με την τοποθεσία. Πιο συγκεκριμένα, όταν η πληροφορία διαβαστεί από το αρχείο /etc/exports, αποθηκεύεται στη δομή *nfsd4_fs_locations* (η δομή που αντιστοιχεί στην θεωρητική *fs_locations* στον εξυπηρετητή). Εν συνεχεία παρεμβαίνουμε πριν την κωδικοποίησή της σε XDR, και έχοντας την πληροφορία του FSN και του κόμβου NSDB, επικοινωνούμε με τον δαίμονα του κόμβου μέσω socket από τον πυρήνα του εξυπηρετητή, και του αποστέλλουμε το FSN. Η πληροφορία της πραγματικής τοποθεσίας αναζητείται και επιστρέφεται στον πυρήνα από τον δαίμονα. Μόλις ολοκληρωθεί η παραλαβή των δεδομένων, ανανεώνουμε την δομή με την πληροφορία που παρελήφθη, την κωδικοποιούμε και την αποστέλλουμε στον πελάτη αφήνοντας το πρωτόκολλο εκεί να την χειριστεί.

Σε πιο τεχνικό επίπεδο, η δομή *nfsd4_fs_locations* περιέχει ένα πίνακα *nfsd4_fs_location* τοποθεσιών, όπου τοποθετούνται ως μεταβλητές ο εξυπηρετητής που διαθέτει το σύνολο αρχείων, οι διευθύνσεις των εξυπηρετητών που διαθέτουν αντίγραφά του, και το μονοπάτι εξαγωγής του συνόλου αρχείων, που είναι κοινό για αυτούς. Σε περίπτωση που υπάρχουν αντίγραφα για τα οποία το μονοπάτι είναι διαφορετικό, αυτά εισάγονται σε παρόμοιες ομάδες στα επόμενα στοιχεία του πίνακα. Ο λόγος που χρειάζονται δύο δομές για την υλοποίηση των παραπάνω έγκειται στον τρόπο με τον οποίο εξετάζεται ο πίνακας από τις συναρτήσεις του πρωτοκόλλου. Πριν την κωδικοποίηση κάθε μεταβλητής *nfsd4_fs_location*, η πληροφορία κάθε FSN περιέχεται στη μεταβλητή του μονοπατιού, ενώ οι διευθύνσεις των κόμβων NSDB βρίσκονται στην μεταβλητή που περιέχει κανονικά τους εξυπηρετητές. Αφού εξετάσουμε όλα τα αντίγραφα που διατίθενται, αποστέλλουμε ξεχωριστά ζεύγη FSN/NSDB στον δαίμονα, ο οποίος απαντά με την πρώτη λειτουργική τοποθεσία, η οποία αποστέλλεται στον πελάτη.



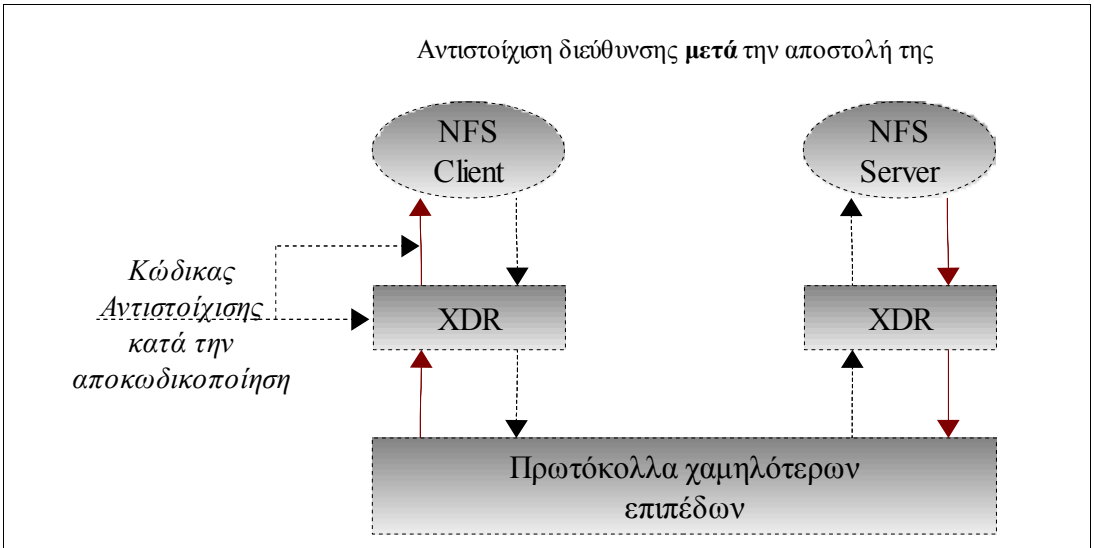
Σχήμα 4.2: Σχηματική αναπαράσταση της λειτουργίας αντιστοίχισης, όταν αυτή εφαρμόζεται στον εξυπηρετητή. Η ροή της πληροφορίας είναι από τον εξυπηρετητή προς τον πελάτη, και η αίτηση για την αντιστοίχιση γίνεται πριν την κωδικοποίηση των δεδομένων στον εξυπηρετητή, με αποτέλεσμα τελικά να κωδικοποιηθεί η πραγματική τοποθεσία του συνόλου αρχείων που αιτείται ο πελάτης.

4.1.2 Αντιστοίχιση στον Πελάτη

Μια εναλλακτική προσέγγιση στο πρόβλημα της αντιστοίχισης είναι αυτή που φαίνεται στο Σχήμα 4.3, όπου το NFSv4 παραμετροποιείται, ώστε ο πελάτης να είναι αυτός που αναλαμβάνει την αντιστοίχιση της ονομασίας με την τοποθεσία. Στον πελάτη, όταν η πληροφορία ληφθεί από τον εξυπηρετητή, αποκωδικοποιείται, και έπειτα αποθηκεύεται στη δομή *nfs4_fs_locations*, όπου και λαμβάνει χώρα η παρέμβαση σε αυτή. Και πάλι, έχουμε την πληροφορία του FSN και του κόμβου NSDB, επικοινωνούμε με τον δαίμονα του κόμβου μέσω socket από τον πυρήνα του πελάτη, και του αποστέλλουμε το FSN. Η πληροφορία της πραγματικής τοποθεσίας αναζητείται και επιστρέφεται στον πυρήνα από τον δαίμονα. Η δομή ανανεώνεται με την πληροφορία που παρελήφθη, και αυτή τη φορά χρησιμοποιείται τοπικά από τους μηχανισμούς του πρωτοκόλλου.

Σε πιο τεχνικό επίπεδο, η δομή *nfs4_fs_locations* περιέχει πολύ περισσότερες μεταβλητές από αυτήν που είδαμε στον εξυπηρετητή, όμως μας απασχολεί κυρίως ο πίνακας τοποθεσιών *nfs4_fs_location*, όπου τοποθετούνται ως μεταβλητές ο εξυπηρετητής που διαθέτει το σύνολο αρχείων, οι διευθύνσεις των εξυπηρετητών που διαθέτουν αντίγραφά του, και το μονοπάτι εξαγωγής του συνόλου αρχείων, που είναι κοινό για αυτούς. Και πάλι, αν υφίστανται αντίγραφα με διαφορετικό μονοπάτι εξαγωγής, έχουν εισαχθεί στα επόμενα στοιχεία του πίνακα. Όταν εξετάζουμε κάθε μεταβλητή *nfs4_fs_location* στον πελάτη, η πληροφορία κάθε FSN περιέχεται στη μεταβλητή του μονοπατιού, ενώ οι διευθύνσεις των κόμβων NSDB βρίσκονται, όπως και πριν, στη μεταβλητή που περιέχει κανονικά τους εξυπηρετητές. Αφού

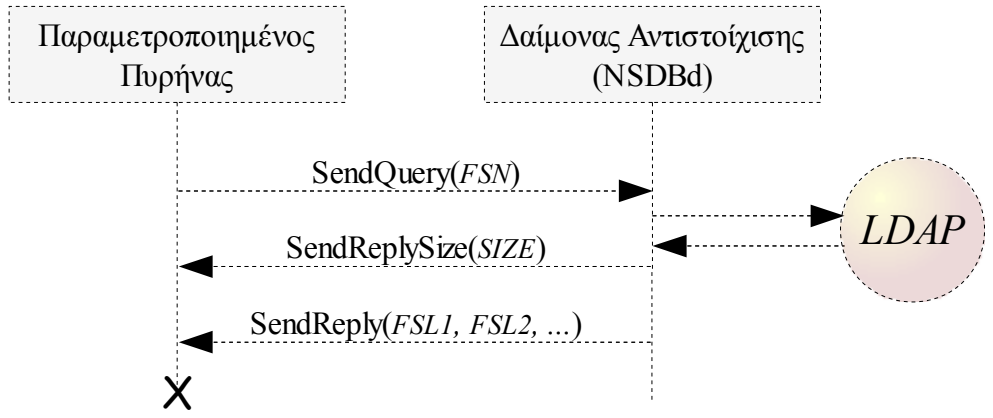
εξετάσουμε όλα τα αντίγραφα που διατίθενται, αποστέλλουμε ξεχωριστά ζεύγη FSN/NSDB στον δαίμονα που απαντά με την πρώτη λειτουργική τοποθεσία, την οποία χρησιμοποιούμε.



Σχήμα 4.3: Σχηματική αναπαράσταση της λειτουργίας αντιστοίχισης, όταν αυτή εφαρμόζεται στον πελάτη. Η ροή της πληροφορίας είναι από τον εξυπηρετητή προς τον πελάτη, και η αίτηση για την αντιστοίχιση γίνεται μετά την κωδικοποίηση των δεδομένων στον πελάτη. Η πληροφορία που αποκωδικοποιείται αποστέλλεται στον δαίμονα αντιστοίχισης, ο οποίος επιστρέφει την πραγματική τοποθεσία του συνόλου αρχείων.

4.2 Δαίμονας Αντιστοίχισης

Ο δαίμονας αντιστοίχισης, όπως φάνηκε και στο Σχήμα 3.1, χρησιμοποιείται μόνο στους κόμβους της υπηρεσίας ονοματοχώρου της ομοσπονδίας. Η ονομασία του είναι ενδεικτική της υπηρεσίας που προσφέρει: Δαίμονας βάσης δεδομένων ονοματοχώρου (*NSDBd* ή *Namespace Database Daemon*). Η λειτουργία που επιτελεί είναι απλή. Δέχεται αιτήσεις αντιστοίχισης μέσω socket και τις υποβάλλει στην τοπική βάση LDAP, στην οποία συνδέεται ανώνυμα. Οι αιτήσεις διασπώνται σε ερωτήματα αναζήτησης, καθώς κάθε ένα μπορεί να περιέχει μόνο ένα FSN και μια διεύθυνση NSDB, αντίθετα με την αίτηση, που όπως είδαμε μπορεί να περιέχει οσαδήποτε αντίγραφα έχουν συμπληρωθεί στην αντίστοιχη εγγραφή του αρχείου ρυθμίσεων του εξυπηρετητή. Τα αποτελέσματα των ερωτημάτων εξετάζονται για να διαπιστωθεί αν έχουν ληφθεί οι τοποθεσίες που ζητήθηκαν. Στην περίπτωση που διαπιστωθεί πως οι τοποθεσίες δεν αποθηκεύονται στην τοπική βάση, τότε γίνεται σύνδεση με τον επόμενο κόμβο που υποδεικνύεται για την αναζήτηση τους κ.ο.κ. μέχρι να βρεθούν οι ζητούμενες πληροφορίες. Όταν βρεθεί η πρώτη έγκυρη τοποθεσία, ο δαίμονας επικοινωνεί και πάλι με τον πυρήνα του πελάτη ή του εξυπηρετητή που τον κάλεσε, και μέσω μιας διεπαφής μηνυμάτων που φαίνεται στο Σχήμα 4.4, του αποστέλλει τις τοποθεσίες, οι οποίες τελικά αποθηκεύονται στην αντίστοιχη δομή. Για να επιτευχθεί καλύτερη απόδοση στην εξυπηρέτηση των αιτήσεων, ο δαίμονας έχει σχεδιαστεί με πολυνηματικό τρόπο, ενώ στην υλοποίηση διατίθεται και μια έκδοσή του που αντικαθιστά τα νήματα με διεργασίες.



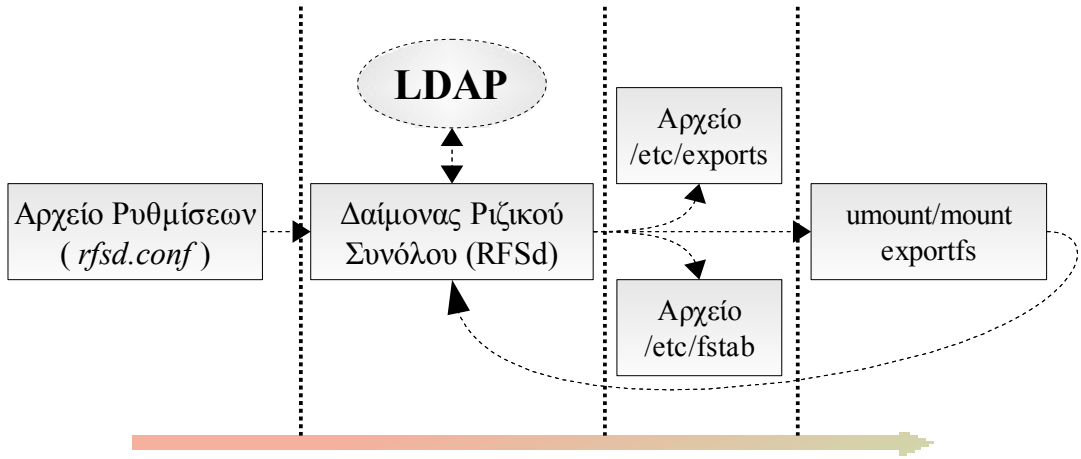
Σχήμα 4.4: Sequence Diagram της επικοινωνίας πυρήνα με το δαίμονα αντιστοίχισης. Η επικοινωνία ξεκινά με τη λήψη ενός μηνύματος SendQuery από τον πυρήνα, το οποίο περιέχει ένα ζεύγος αναγνωριστικού FSN, και μίας διεύθυνσης NSDB. Ο δαίμονας αναζητά το FSN στην τοπική βάση LDAP, και αν η αναζήτηση αποτύχει εξετάζει την επόμενη βάση, στην οποία παραπέμπεται από τις τοπικές πληροφορίες. Όταν η αναζήτηση λήξει, αποστέλλεται απάντηση SendReplySize στον πυρήνα είτε με αποτυχία, είτε με το μέγεθος των τοποθεσιών που βρέθηκαν και πρόκειται να σταλούν. Στην δεύτερη περίπτωση, ο πυρήνας εκχωρεί τον αντίστοιχο χώρο στη δομή του, και εν συνεχεία την συμπληρώνει με την πληροφορία SendReply που ακολουθεί.

4.3 Δαίμονας Ριζικού Συνόλου Αρχείων

Η υλοποίηση ενός ομοσπονδιακού συστήματος δεν θα μπορούσε να ολοκληρωθεί χωρίς την ύπαρξη μιας διεπαφής η οποία να διαχειρίζεται το ριζικό σύνολο αρχείων, φροντίζοντας να ενημερώνει τους ριζικούς κόμβους για τη δομή του, ώστε οι πελάτες που απευθύνονται σε αυτούς να έχουν μια συνεπή εικόνα του ονοματοχώρου. Το όνομα του δαίμονα αυτού είναι και πάλι χαρακτηριστικό της λειτουργίας του: Δαίμονας ριζικού συνόλου αρχείων (*Root fileset daemon* ή *RFSd*). Όπως προείπαμε, ο δαίμονας αυτός δεν ενημερώνει τον πελάτη για αλλαγές στη δομή του ριζικού συνόλου, και έτσι απαιτείται από τον πελάτη να αναλαμβάνει την επαναπροσάρτηση του ριζικού συνόλου κατά απαίτηση όταν αντιλαμβάνεται αλλαγές. Με αυτό τον τρόπο αποφεύγονται εμπλοκές με εργασίες που γίνονται στον ονοματοχώρο από τοπικούς χρήστες. Εντούτοις, αυτό δεν είναι απαραίτητο να γίνει άμεσα, καθώς κάθε σύνολο αρχείων αποτελεί αυτόνομη οντότητα της ομοσπονδίας, με αποτέλεσμα ο πελάτης να δύναται να εκτελέσει τοπικές εργασίες στο σύνολο ακόμα και αν η λογική του θέση έχει αλλάξει.

Ρύθμιση δαίμονα. Η ρύθμιση του δαίμονα γίνεται στο αρχείο /etc/rfsd.conf, στο οποίο ορίζονται τέσσερις παράμετροι, εξαρτώμενες από τον ριζικό εξυπηρετητή όπου εκτελείται ο δαίμονας. Ο διαχειριστής του συστήματος πρέπει να ενημερώσει το αρχείο ορίζοντας τον NSDB κόμβο από όπου πρόκειται να αντληθούν οι πληροφορίες του ριζικού συνόλου, το δίκτυο ή τις διευθύνσεις που επιτρέπεται να προσαρτούν τα εξαγόμενα δεδομένα, και σε περίπτωση που επιλεγεί δίκτυο, τότε πρέπει να οριστεί και η μάσκα του υποδικτύου. Τέλος, ορίζεται και το χρονικό διάστημα ανά το οποίο εκτελείται η ενημέρωση του ριζικού συνόλου.

Λειτουργία δαίμονα. Η λειτουργία του δαίμονα περιορίζεται σε προφανή βήματα, η εκτέλεση των οποίων αρκεί για την εξαγωγή του ενημερωμένου συνόλου. Όπως φαίνεται και στο σχήμα 4.5, ο δαίμονας εκτελείται σε τακτές επαναλήψεις, όπως ορίζεται στο αρχείο ρυθμίσεων, σε κάθε μια από τις οποίες επικοινωνεί με τον ενδεδειγμένο NSDB κόμβο και λαμβάνει τα δεδομένα που αφορούν το ριζικό σύνολο. Στη συνέχεια τα συνθέτει, και κάνει τις απαραίτητες αλλαγές στα αρχεία /etc/exports και /etc/fstab, ώστε να εξάγουν σωστά τους φακέλους και τα σημεία προσάρτησης του συνόλου. Τέλος, καλεί τις απαραίτητες υπηρεσίες ώστε να ενημερώσουν τον εξυπηρετητή για τις νέες ρυθμίσεις.



Σχήμα 4.5: Σχηματική αναπαράσταση της λειτουργίας του δαίμονα ριζικού συνόλου. Κατά την εκκίνησή του, ο δαίμονας ενημερώνεται για τις ρυθμίσεις λειτουργίας του από το αρχείο /etc/rfsd.conf του συστήματος. Στη συνέχεια μπαίνει σε έναν επαναληπτικό βρόγχο, κάθε επανάληψη του οποίου λαμβάνει χώρα σε τακτά χρονικά διαστήματα που ορίζονται στο αρχείο ρυθμίσεων. Σε κάθε βήμα γίνεται επικοινωνία με τον ενδεδειγμένο κόμβο NSDB, λαμβάνονται οι πληροφορίες του ριζικού συνόλου και ακολουθεί η επεξεργασία τους. Από τα δεδομένα που προκύπτουν συντίθενται νέα αρχεία /etc/exports και /etc/fstab, τα οποία φορτώνονται στο τέλος της επανάληψης, με την κλήση των απαραίτητων υπηρεσιών.

5 *Πειραματικά Αποτελέσματα*

"πείραμα" (το) ουσ. *δοκιμή, απόπειρα* || (ειδ.) *τεχνητή παραγωγή φυσικών, φυσιολογικών ή ψυχικών φαινομένων για μελέτη*

Στο κεφάλαιο αυτό περιγράφονται τα πειράματα που εκτελέστηκαν στην υλοποίηση που αναλύθηκε, με σκοπό την εξαγωγή συμπερασμάτων για την απόδοση ενός ομοσπονδιακού συστήματος, και τη συμμετοχή που έχει σε αυτή ο μηχανισμός αντιστοίχισης. Αρχικά, γίνεται μια εισαγωγή στην τεχνολογία των εικονικών μηχανών, η οποία χρησιμοποιήθηκε για να καλύψει τις απαιτήσεις του πειραματικού συστήματος σε υλικό, καθώς δεν θα μπορούσαν να καλυφθούν διαφορετικά. Στη συνέχεια του κεφαλαίου, γίνεται άμεση σύγκριση μεταξύ των υλοποιήσεων που εφαρμόζουν την αντιστοίχιση είτε στον εξυπηρετητή, είτε στον πελάτη, εξετάζοντας δύο σενάρια: τη διάσχιση του ονοματοχώρου και την επεξεργασία δεδομένων που βρίσκονται διάσπαρτα στην ομοσπονδία. Χρησιμοποιούμε έναν ή δύο πελάτες, οι οποίοι ξεκινούν πάντοτε ταυτόχρονα την ίδια ακριβώς λειτουργία, και παρατηρούμε των τρόπο που ανταποκρίνεται το σύστημα ως προς διάφορα μεγέθη.

5.1 Χρήση περιβάλλοντος προσομοίωσης

Πειραματικό σύστημα. Το σύστημα που χρησιμοποιείται για τα πειράματα έχει την απλούστερη δυνατή δομή που θα μπορούσε να θεωρηθεί ότι αποτελεί μια ομοσπονδία. Αποτελείται από έναν κόμβο NSDB, έναν ριζικό εξυπηρετητή, έναν εξυπηρετητή αρχείων και τους πελάτες, οι οποίοι εδώ δεν θα ξεπεράσουν τους δύο. Παρόλο που αυτή είναι η πιο απλή δομή ενός ομοσπονδιακού συστήματος, παρατηρούμε πως για την πειραματική υλοποίησή

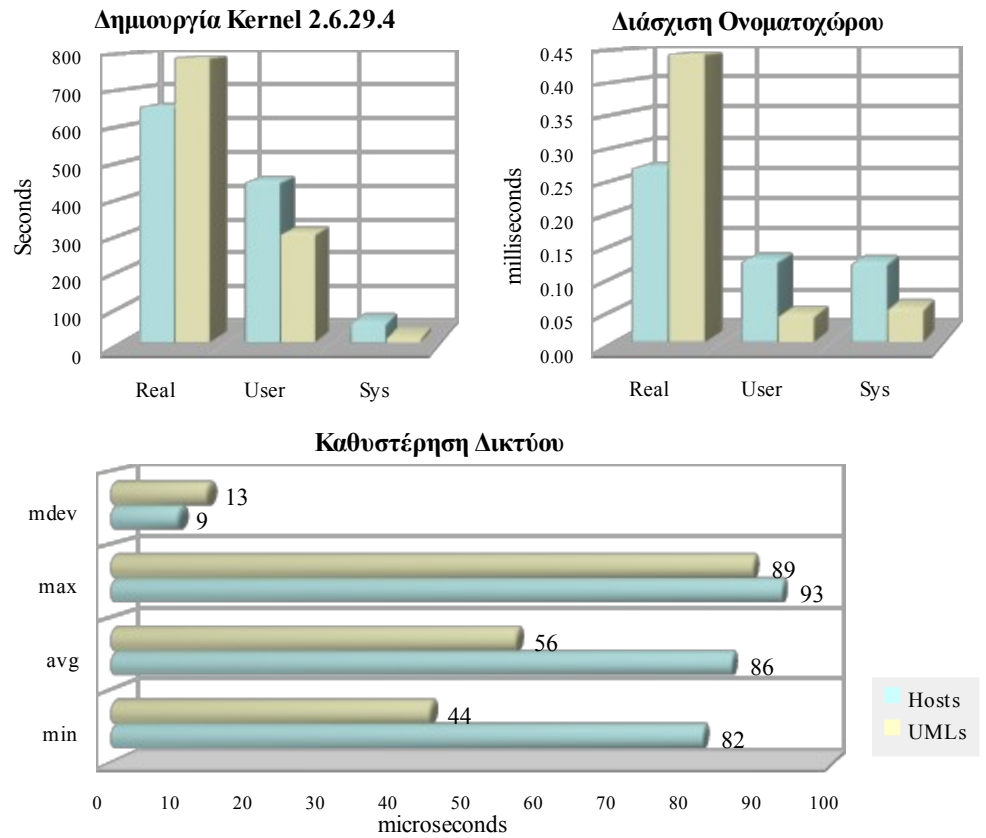
της, υπάρχει η απαίτηση τουλάχιστον πέντε ανεξάρτητων κόμβων, κάθε ένας από τους οποίους απαιτείται να λειτουργεί με ειδικά παραμετροποιημένο πυρήνα, που καθορίζεται από το πείραμα που εκτελείται.

Χρήση εικονικών μηχανών. Για λόγους απλότητας, η υλοποίηση του συστήματος όπου εκτελέστηκαν τα πειράματα, έγινε με τη βοήθεια εικονικών μηχανών UML. Η τεχνολογία UML [Dike06], εν ολίγοις, λειτουργεί σε περιβάλλον linux, και υλοποιεί ένα kernel στον χώρο χρήστη. Πάνω στον πυρήνα αυτόν, μπορεί να χτιστεί ένα εικονικό λειτουργικό σύστημα linux, το οποίο λειτουργεί όμως σαν διεργασία στο περιβάλλον του λειτουργικού συστήματος που το φιλοξενεί. Η UML δεν έχει πρόσβαση στο πραγματικό υλικό και στους πόρους του υπολογιστή, τουλάχιστον χωρίς τη συγκατάθεση του χρήστη. Το υλικό που αντιλαμβάνεται είναι εικονικό, ενώ ακόμα και οι σκληροί δίσκοι του συστήματος, αποτελούνται από απλά αρχεία στο σύστημα που φιλοξενεί την εικονική μηχανή. Ως επέκταση αυτού, η UML μας δίνει την δυνατότητα να δημιουργήσουμε εικονικό υλικό που δεν υφίσταται, με σκοπό τη δοκιμή διαφορετικών ρυθμίσεων του συστήματος. Στα πειράματα που εκτελούμε εδώ, κάθε κόμβος UML απαιτείται να συμπεριφέρεται ως ανεξάρτητος κόμβος του συστήματος, και έτσι είναι όλοι τους συνδεδεμένοι στο διαδίκτυο, έχουν διαφορετικές διευθύνσεις IP, ενώ χρησιμοποιούν ο καθένας έναν επεξεργαστή από το αρχικό σύστημα, διαφορετικό κομμάτι εικονικής μνήμης RAM και αρχείο σκληρού δίσκου.

Απόδοση εικονικών μηχανών. Το ερώτημα που ανακύπτει με την επιλογή εικονικών μηχανών είναι προφανές, και αφορά την εγκυρότητα των πειραματικών αποτελεσμάτων που προκύπτουν με τη χρήση τους. Στο σχήμα 5.1, φαίνονται τα αποτελέσματα των πειραμάτων που εκτελέστηκαν για την σύγκριση τους με πραγματικές συνθήκες. Από αυτά, προκύπτει πως η δημιουργία πυρήνα στο εικονικό σύστημα απαιτεί σχεδόν 20% περισσότερο χρόνο από ότι στο λειτουργικό σύστημα που το φιλοξενεί. Η διάσχιση ονοματοχώρου απαιτεί περίπου 40% περισσότερο χρόνο, ενώ η διάδοση δεδομένων μέσω δικτύου γίνεται 35% γρηγορότερα. Παρόλο που οι καθυστερήσεις αυτές είναι αναμενόμενες, δεν θεωρούμε ότι είναι ικανές να καταστήσουν τα αποτελέσματα των πειραμάτων ανακριβή, καθώς η χρήση τους περιορίζεται στη σύγκριση περιπτώσεων, οι οποίες εξετάζονται όλες στο εικονικό περιβάλλον. Επομένως, ενδιαφερόμαστε μόνο για την τάση που παρουσιάζεται στο σύστημα όσο αλλάζουμε τις συνθήκες του πειράματος. Δεδομένου ότι κάθε εικονική μηχανή καταλαμβάνει μόνο ένα κομμάτι του πραγματικού υλικού του κόμβου, αποκλείονται πιθανές καθυστερήσεις που θα μπορούσαν να προκύψουν από τον διαμοιρασμό του υλικού σε πολλαπλά μηχανήματα.

Σύστημα εκτέλεσης πειραμάτων. Το δίκτυο εκτέλεσης των πειραμάτων αποτελείται από δύο κόμβους, καθένας από τους οποίους διαθέτει τέσσερις επεξεργαστές Intel Xeon® E5345,

με χρονισμό 2.33 GHz, 2GB μνήμης RAM και δύο σκληρούς δίσκους Western Digital® WD2500YS-70S των 7200 RPM, χωρητικότητας 250GB ο καθένας. Στον ένα από αυτούς τους κόμβους υλοποιείται ο κόμβος NSDB του συστήματος, ενώ στον άλλο τοποθετήθηκε ο ριζικός εξυπηρετητής, καθώς και ο εξυπηρετητής αρχείων. Επίσης, χρησιμοποιήθηκε ένας κόμβος με το ίδιο μοντέλο επεξεργαστή, 3GB μνήμης RAM και δύο σκληρούς δίσκους Hewlett Packard® GB0500EAJFH και GB0500C4413 των 7200 RPM, χωρητικότητας 500GB ο καθένας, στον οποίον τοποθετήθηκαν οι κόμβοι με τους πελάτες του συστήματος.



Σχήμα 5.1: Σύγκριση απόδοσης δικτύου εικονικών μηχανών σε αντιδιαστολή με το πραγματικό δίκτυο, στη δημιουργία πυρήνα (`make defconfig ARCH=x86 && time make -j1 ARCH=x86`), διάσχιση πρότυπου ονοματοχώρου (`ls -R linux-2.6.29.4/`) και μέτρηση δικτυακών καθυστερήσεων (`ping -c10 -s64`).

5.2 Κόστος διάσχισης ομοσπονδιακού ονοματοχώρου

Σκοπός πειράματος. Το πείραμα έχει ως σκοπό την αξιολόγηση του φόρτου με τον οποίο επιβαρύνεται το σύστημα από τη χρήση του μηχανισμού αντιστοίχισης. Ο μηχανισμός δεν σχετίζεται με τις αιτήσεις που κάνει ο πελάτης για δεδομένα, και έτσι χρησιμοποιούμε την λειτουργία `ls` του Unix, η οποία στην αναδρομική της μορφή, διατρέχει ένα δένδρο καταλόγων, και τυπώνει στην οθόνη τα περιεχόμενα των φακέλων που το αποτελούν. Στην πραγματικότητα η λειτουργία δεν εξετάζει τα περιεχόμενα των αρχείων που διαπερνά, παρά μόνο τα μεταδεδομένα που τα περιγράφουν. Έτσι, ο χρόνος εκτέλεσής της, αλλά και τα

δεδομένα που ανταλλάσσονται θεωρούνται αρκετά μικρά μεγέθη, ώστε να μπορέσουμε να εκτιμήσουμε εύκολα το κόστος της αντιστοίχισης, η οποία απαιτεί εξ' ορισμού λίγο χρόνο για την εκτέλεσή της.

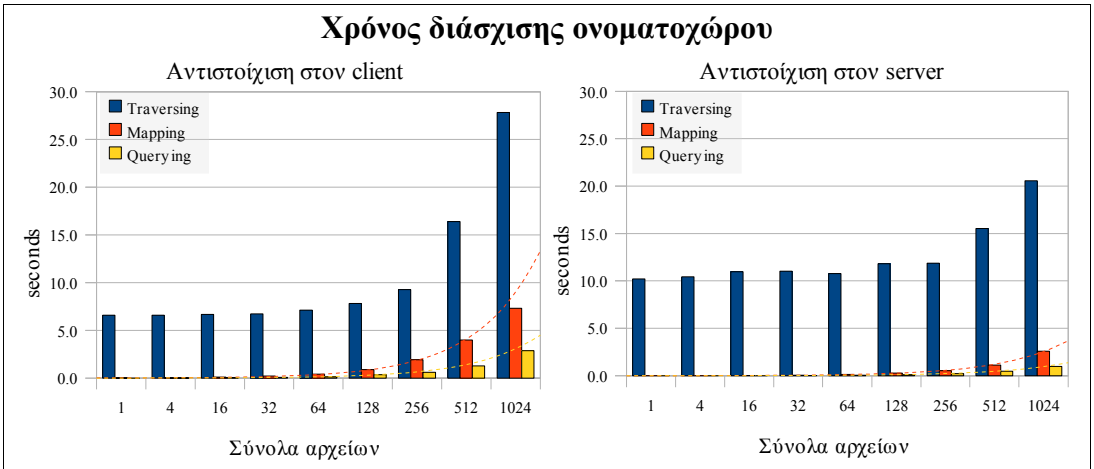
Περιγραφή πειράματος. Το δένδρο καταλόγων που χρησιμοποιούμε ως ονοματοχώρο στα πειράματα που ακολουθούν, είναι αυτό που προκύπτει από την αποσυμπίεση του πηγαίου κώδικα του πυρήνα 2.6.29.4 του linux [Kernel29.4]. Ο συνολικός αριθμός φακέλων που περιέχονται σε αυτό είναι 1675, και έτσι ο μέγιστος αριθμός διαφορετικών συνόλων αρχείων που μπορούμε να δημιουργήσουμε είναι ισάριθμος. Εδώ, αλλά και στα επόμενα πειράματα, θεωρούμε ότι το όριο των 1024 συνόλων είναι αρκετό, και στις επαναλήψεις κάθε πειράματος αυξάνουμε τον αριθμό τους εκθετικά, παρατηρώντας κάθε φορά την μεταβολή που απαιτείται για τον χρόνο εκτέλεσης της λειτουργίας. Εντούτοις, κάθε φορά που αλλάζει ο αριθμός των συνόλων αρχείων απαιτείται ρύθμιση των αρχείων `exports` και `fstab` του εξυπηρετητή, καθώς και ενημέρωση της βάσης LDAP του NSDB. Επειδή παράλληλα πρέπει να αλλάζει και το kernel ορισμένων κόμβων ανάλογα με το πείραμα που εκτελείται, η ρύθμιση του συστήματος απαιτεί την επανεκκίνηση των μηχανών για την απόκτηση κάθε δείγματος, με αποτέλεσμα να υπάρχει περιορισμός στον αριθμό των δειγμάτων που μπορούν να ληφθούν στο χρονικό διάστημα που εκπονείται η εργασία. Έτσι, επιλέγουμε να κάνουμε 9 διαφορετικές δειγματοληψίες με έναν και δύο πελάτες για κάθε υλοποίηση, χρησιμοποιώντας από 1 έως 1024 σύνολα αρχείων για κάθε πείραμα, αυξάνοντας εκθετικά τον αριθμό τους, και αφαιρώντας τις περιπτώσεις για 2 και 8 σύνολα αρχείων, καθώς τα αποτελέσματά από 1 μέχρι 16 σύνολα δεν παρουσιάζουν μεγάλες μεταβολές. Κάθε επανάληψη εκτελέστηκε 3 φορές, αλλά κυρίως για τυπικούς λόγους, καθώς δεν υπήρχε απόκλιση μεγαλύτερη του 1% στα αποτελέσματα, που επαρκεί για την εξαγωγή συμπερασμάτων.

5.2.1 Διάσχιση με έναν πελάτη

Αρχικά, χρησιμοποιούμε το σύστημα με την διάταξη που περιγράφηκε, με μονάχα έναν πελάτη, ώστε να αντιληφθούμε το γενικότερο κόστος της αντιστοίχισης και να μπορέσουμε να κάνουμε μια απευθείας σύγκριση μεταξύ των δύο υλοποιήσεων που αντιστοιχίζουν την πραγματική και την εικονική θέση των δεδομένων είτε στον πελάτη, είτε στον εξυπηρετητή.

Χρόνος διάσχισης ονοματοχώρου. Όπως φαίνεται στο σχήμα 5.2.1, η αύξηση του χρόνου που απαιτείται για να ολοκληρωθεί η λειτουργία αυξάνεται ανάλογα με τον αριθμό των συνόλων αρχείων του ονοματοχώρου. Ο χρόνος διάσχισης (*traversing time*) που φαίνεται στο σχήμα, περιλαμβάνει και τον συνολικό χρόνο αντιστοίχισης (*mapping time*), ο οποίος καταγράφει το χρονικό διάστημα από τη στιγμή που ο πυρήνας ζητά την πληροφορία, μέχρι τη στιγμή που την λαμβάνει, συμπεριλαμβάνοντας το τρίτο μέγεθος (*querying time*), που

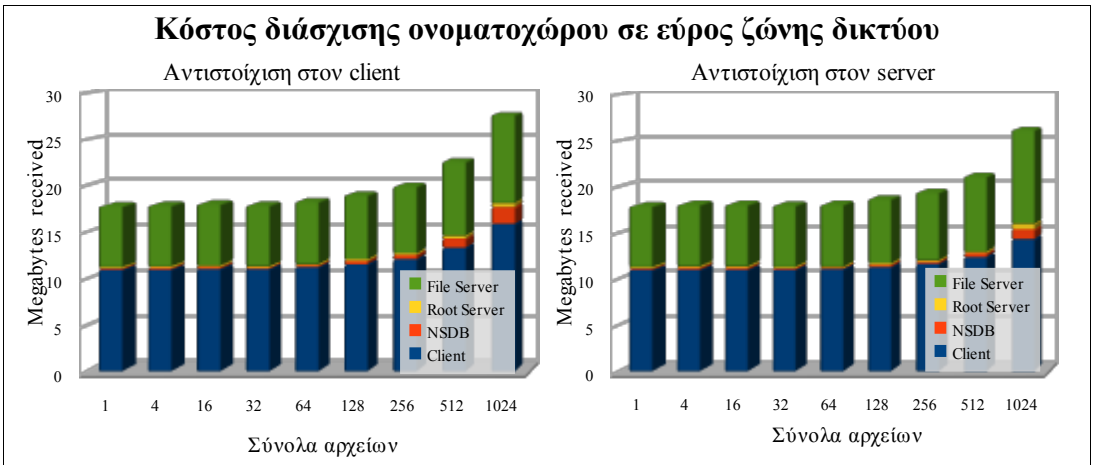
είναι ο χρόνος που χρειάζεται ο δαίμονας αντιστοίχισης για την επερώτηση στην βάση LDAP που περιέχει τις ζητούμενες πληροφορίες. Παρατηρούμε από το σχήμα, πως όταν η αντιστοίχιση γίνεται στον πελάτη, ο χρόνος που χρειάζεται για αυτή αρχικά καταλαμβάνει το 0.19% του χρόνου που απαιτείται για τη λειτουργία, και τελικά καταλήγει να αυξάνεται έως και 26.31%, ενώ όταν η αντιστοίχιση γίνεται στον εξυπηρετητή, τα ποσοστά κυμαίνονται από 0.06% έως και 12.65%. Η μείωση αυτή είναι αναμενόμενη, καθώς ο χρόνος που θεωρούμε για την αντιστοίχιση, περιλαμβάνει την ανταλλαγή όλων των απαιτούμενων μηνυμάτων που περιγράφηκαν στο κεφάλαιο 4. Έτσι, στην περίπτωση του πελάτη παρατηρείται μεγαλύτερη καθυστέρηση, καθώς είναι βεβαρημένος με την επεξεργασία της πληροφορίας που φτάνει για την ίδια τη λειτουργία, με αποτέλεσμα να καθυστερεί η επικοινωνία του με το δαίμονα. Αυτό δεν συμβαίνει και στην περίπτωση του εξυπηρετητή, καθώς ο τελευταίος δεν επιβαρύνεται με κάποια άλλη λειτουργία πέραν της αντιστοίχισης, τουλάχιστον στο παρόν πείραμα.



Σχήμα 5.2.1: Σύγκριση χρόνου διάσχισης του δένδρου καταλόγων του kernel 2.6.29.4, με την αντιστοίχιση να γίνεται στον client και στον server. Παρατηρούμε ότι η αύξηση του χρόνου είναι ανάλογη αυτής των συνόλων αρχείων, ενώ η αντιστοίχιση καθυστερεί περισσότερο στον πελάτη, λόγω της επιβάρυνσης που προκαλεί η επεξεργασία των δεδομένων που λαμβάνονται για τη λειτουργία από τον εξυπηρετητή αρχείων.

Μια άλλη παρατήρηση στο σχήμα 5.2.1, είναι η αλλαγή της καθυστέρησης μεταξύ των δειγμάτων, η οποία δεν περιλαμβάνει μόνο την επιπλέον καθυστέρηση που παρατηρείται στην αντιστοίχιση. Ο επιπλέον χρόνος οφείλεται στο πρωτόκολλο NFSv4, και συγκεκριμένα στην επίλυση των αναφορών (*referrals*) που αναγκάζεται να εκτελέσει, λόγω της διάσχισης διαφορετικών συνόλων αρχείων. Ο χρόνος αυτός παρατηρούμε πως είναι εξίσου σημαντικός και στις δύο υλοποιήσεις. Σημαντικό, όμως, είναι να παρατηρηθεί και ο χρόνος που απαιτείται συγκριτικά από τις δύο υλοποιήσεις, για την ολοκλήρωση της λειτουργίας. Ενώ η αντιστοίχιση στον πελάτη φαίνεται να έχει αρχικά χρονικό προβάδισμα, το γεγονός πως τα δεδομένα της αντιστοίχισης επιβαρύνουν τον εξυπηρετητή, δείχνει να επιτρέπει στην αντιστοίχιση από τον εξυπηρετητή να υπερισχύσει για μεγάλο αριθμό συνόλων αρχείων. Στην πράξη, αν οι εξυπηρετητές που φιλοξενούν τα σύνολα είναι απομακρυσμένοι μεταξύ

τους η καθυστέρηση αυτή αυξάνεται, όμως το γεγονός αυτό οφείλεται στο NFSv4, το οποίο δέχεται πλήγμα στην απόδοση και χωρίς το κόστος αντιστοίχισης, που αποδείχθηκε πως δεν είναι απαγορευτικό ως προς τη χρήση της.

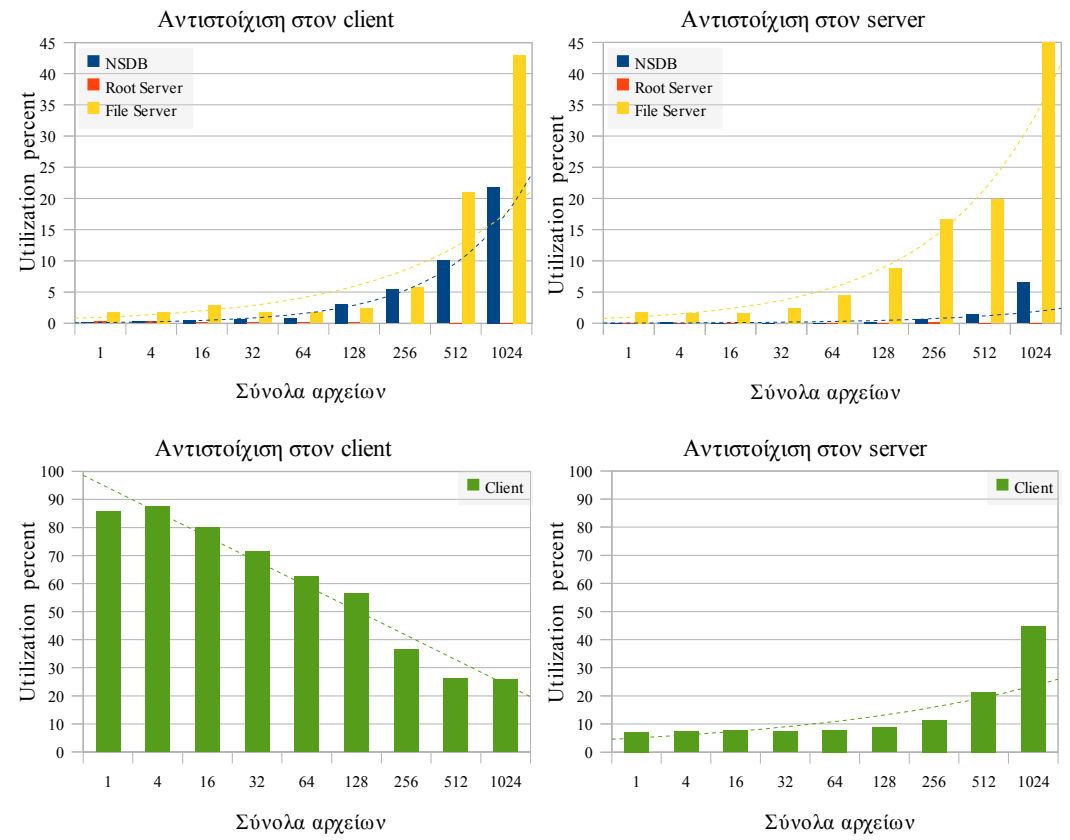


Σχήμα 5.2.2: Σύγκριση κόστους διάσχισης δένδρου καταλόγων του kernel 2.6.29.4 σε εύρος ζώνης, με την αντιστοίχιση να γίνεται στον client και στον server. Παρατηρούμε ότι τα ανταλλασσόμενα δεδομένα αυξάνονται με τον αριθμό των συνόλων αρχείων, ενώ η αντιστοίχιση δεν δημιουργεί ιδιαίτερη επιβάρυνση στον κόμβο NSDB, σε σχέση με αυτή που δέχονται ο εξυπηρετητής αρχείων και ο πελάτης από τη διάσχιση.

Κόστος διάσχισης ονοματοχώρου. Το κόστος της διάσχισης του ονοματοχώρου μπορεί να αναλυθεί με την βοήθεια δύο σημαντικών μεγεθών. Αφενός θα εξετάσουμε τα δεδομένα που ανταλλάχθηκαν μέσω δικτύου κατά την εκτέλεση της λειτουργίας, και αφετέρου θα δούμε πόσο επηρεάζεται η επεξεργαστική ισχύς που απαιτείται, από την αύξηση του αριθμού των συνόλων αρχείων. Στο σχήμα 5.2.2 φαίνεται ο όγκος των δεδομένων που ελήφθησαν κατά τη διάρκεια των πειραμάτων από τους κόμβους του συστήματος. Είναι εμφανές ότι ο κόμβος NSDB και ο εξυπηρετητής ρίζας δεν λαμβάνουν σημαντικό μέγεθος δεδομένων, παρόλο που για τον πρώτο το μέγεθος αυξάνεται με την αύξηση του αριθμού των αιτήσεων. Η τάξη μεγέθους αυτή, που παρουσιάζεται με κόκκινο χρώμα στο σχήμα, αποτελεί και τον όγκο των δεδομένων που παράγεται λόγω των αιτήσεων αντιστοίχισης, ενώ ανάλογος είναι και ο όγκος δεδομένων με τον οποίο επιβαρύνεται ο κόμβος που αναλαμβάνει την αντιστοίχιση. Εκ πρώτης όψεως, τα δεδομένα στον πελάτη και τον εξυπηρετητή αρχείων δεν φαίνεται να διαφέρουν πολύ στις δύο περιπτώσεις. Όμως, παρόλο που ο αριθμός αιτήσεων είναι ο ίδιος, και το μικρό μέγεθος δεδομένων που τις εκφράζει χάνεται στην ποσότητα των δεδομένων που απαιτούνται για την εκτέλεση της λειτουργίας, μπορούμε να παρατηρήσουμε πως ο πελάτης λαμβάνει λιγότερα δεδομένα όταν αναλαμβάνει την αντιστοίχιση ο εξυπηρετητής, κάτι που γίνεται ιδιαίτερα εμφανές στο σχήμα, για μεγάλο αριθμό αιτήσεων.

Η επεξεργαστική ισχύς που απαιτείται για την εκτέλεση της διάσχισης παρουσιάζεται στο σχήμα 5.2.3. Είναι εμφανές, πως η αύξηση του αριθμού των συνόλων αρχείων έχει επίπτωση

στον επεξεργαστικό φόρτο των κόμβων του συστήματος. Συγκεκριμένα, ο ριζικός εξυπηρετητής δεν επηρεάζεται, καθώς δέχεται μόνο μια αίτηση για τον εντοπισμό της ρίζας του ονοματοχώρου, ενώ ο εξυπηρετητής αρχείων και ο κόμβος NSDB παρουσιάζουν αύξηση στην επεξεργαστική ισχύ, ανάλογη με τον αριθμό αιτήσεων που διαχειρίζονται. Ο ρυθμός της αύξησης αυτής είναι μικρότερος στον NSDB όταν η αντιστοίχιση γίνεται στον εξυπηρετητή, καθώς όπως ειπώθηκε, η έλλειψη καθυστέρησης που συνεπάγεται η διάσχιση, επιτρέπει στον εξυπηρετητή αρχείων να επιλύει τις αντιστοιχίσεις γρηγορότερα.



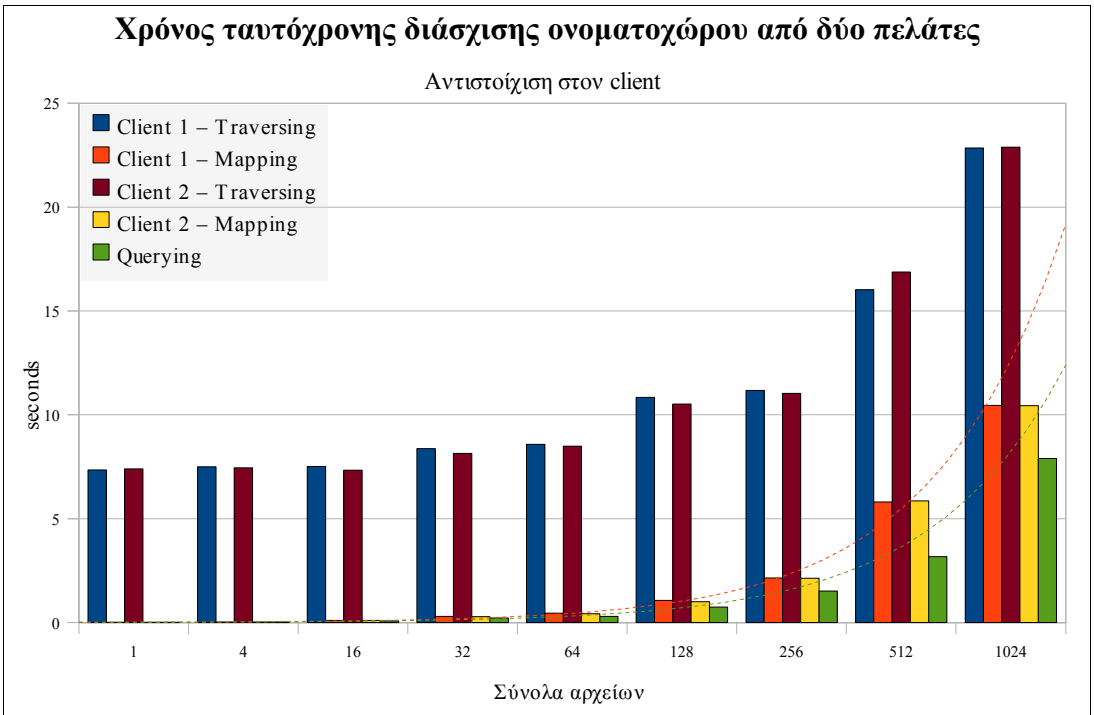
Σχήμα 5.2.3: Σύγκριση επεξεργαστικού κόστους διάσχισης δένδρου καταλόγων του kernel 2.6.29.4, με την αντιστοίχιση να γίνεται στον client και στον server. Παρατηρούμε ότι η αύξηση του επεξεργαστικού φόρτου είναι ανάλογη αυτής των συνόλων αρχείων, ενώ όταν η αντιστοίχιση γίνεται μαζί με την επεξεργασία των υπόλοιπων δεδομένων στον πελάτη, η ανταλλαγή μηνυμάτων με τον κόμβο NSDB καθυστερεί, με αποτέλεσμα να δημιουργείται μεγαλύτερη επεξεργαστική επιβάρυνση στον τελευταίο. Επίσης, ο πελάτης φαίνεται να αδρανεί περισσότερο, όσο αυξάνονται τα δεδομένα που δέχεται από διαφορετικά σύνολα αρχείων.

Αξιοσημείωτο είναι το φαινόμενο που παρατηρείται στον πελάτη. Όταν η αντιστοίχιση γίνεται εκεί, τότε παρατηρούμε μεγάλο επεξεργαστικό φόρτο, ο οποίος μειώνεται όσο αυξάνουμε τα σύνολα αρχείων. Αυτό από μόνο του φαίνεται λογικό, καθώς ο πελάτης αναμένει περισσότερα δεδομένα από το δίκτυο, με αποτέλεσμα να καθυστερεί η επεξεργασία τους και να αυξάνεται ο απαιτούμενος χρόνος, με αποτέλεσμα να μειώνεται ο μέσος όρος της επεξεργαστικής ισχύος που χρησιμοποιείται. Αντίθετα, όταν η αντιστοίχιση γίνεται στον

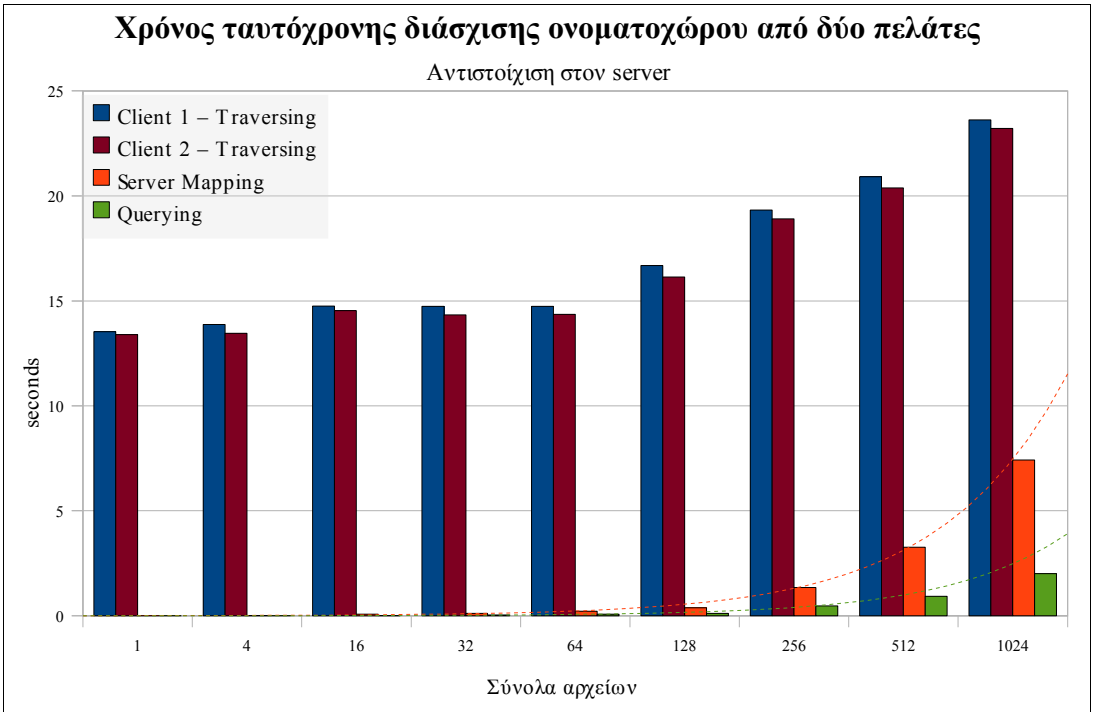
εξυπηρετητή, τότε παρατηρούμε ότι ο επεξεργαστής λειτουργεί ελάχιστα. Ο λόγος για αυτό μπορεί να προκύψει από εξέταση του σχήματος 5.2.1, όπου είναι εμφανές ότι απαιτείται περισσότερος χρόνος για την εκτέλεση της λειτουργίας, λόγω καθυστερήσεων στη μεταφορά των δεδομένων προς επεξεργασία μέσω δικτύου. Σε όλο αυτό το επιπλέον χρονικό διάστημα, που μπορεί να φτάσει και το ήμισυ του χρόνου λειτουργίας, ο επεξεργαστής μένει αδρανής, με αποτέλεσμα να μειώνεται ο μέσος όρος χρήσης του, όπως φαίνεται στο σχήμα.

5.2.2 Ταυτόχρονη διάσχιση με δύο πελάτες

Μια αναμενόμενη παραλλαγή του πειράματος είναι η χρήση πολλαπλών πελατών ώστε να εκτιμηθεί η κλιμάκωση που παρατηρείται στον προκαθορισμένο φόρτο του συστήματος. Καθώς ένα ομοσπονδιακό σύστημα οφείλει να έχει μεγάλες δυνατότητες κλιμάκωσης, οι ταυτόχρονες αιτήσεις πολλαπλών πελατών για τον ίδιο πόρο είναι κάτι αναμενόμενο. Επίσης, το NFSv4 έχει δυνατότητες προσωρινής αποθήκευσης, ώστε η επανάληψη μιας αντιστοίχισης από τον ίδιο πελάτη σε εύλογο χρονικό διάστημα, να απαιτεί μόνο την ανάκτηση τοπικής πληροφορίας. Περαιτέρω, αν την αντιστοίχιση αναλαμβάνει ο εξυπηρετητής, η πληροφορία διατίθεται και σε επόμενους πελάτες που πρόκειται να την αναζητήσουν, με το κόστος που είδαμε, αλλά χωρίς να απαιτείται το κόστος επανάληψης της αναζήτησής της. Λόγω των παραπάνω, έχει νόημα να εξετάσουμε μόνο την περίπτωση παράλληλης αποστολής αιτήσεων.



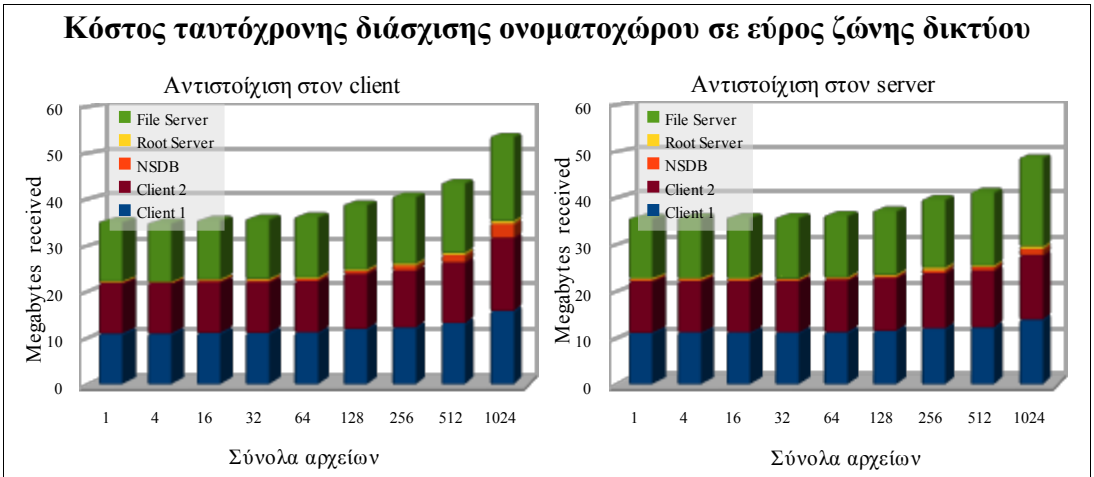
Σχήμα 5.2.4: Μεταβολή του απαιτούμενου χρόνου για τη διάσχιση δένδρου καταλόγων του kernel 2.6.29.4, με την αντιστοίχιση να γίνεται στον client. Παρατηρούμε ότι η αύξηση του χρόνου είναι ανάλογη αυτής των συνόλων αρχείων, ενώ οι χρόνοι ταυτίζονται και για τους δύο πελάτες, αν και είναι αντιληπτή η καθυστέρηση που δημιουργείται από την παράλληλη εκτέλεση παράλληλων αντιστοιχίσεων στον κόμβο NSDB.



Σχήμα 5.2.5: Μεταβολή του απαιτούμενου χρόνου για τη διάσχιση δένδρου καταλόγων του kernel 2.6.29.4, με την αντιστοίχιση να γίνεται στον server. Παρατηρούμε ότι η αύξηση του χρόνου είναι ανάλογη αυτής των συνόλων αρχείων, ενώ οι χρόνοι ταυτίζονται και για τους δύο πελάτες. Αν και ο χρόνος αντιστοίχισης έχει μειωθεί δραματικά, η μεταφορά των μεταδεδομένων από τον server καθυστερεί τη λειτουργία.

Χρόνος ταυτόχρονης διάσχισης. Στο σχήμα 5.2.4, οι λειτουργίες αντιστοίχισης στους πελάτες έχουν εκτελεστεί παράλληλα, και παρατηρούμε αφενός διαφορές στον απαιτούμενο χρόνο σε σύγκριση με αυτόν του σχήματος 5.2.1, και αφετέρου ομοιογένεια στους χρόνους που καταλαμβάνει τόσο η εκτέλεση της λειτουργίας, όσο και η αντιστοίχιση. Μεταξύ των δύο σχημάτων παρατηρούμε ότι η καθυστέρηση στον απαιτούμενο χρόνο αντιστοίχισης είναι έντονη όταν δύο πελάτες αιτούνται τον ίδιο όγκο πληροφορίας από τον δαίμονα, αλλά παρόλο που οι αιτήσεις που τίθενται είναι διπλάσιες, ο χρόνος αυξάνεται μόνο κατά 32%, καθώς το μεγαλύτερο μέρος του αρχικού χρόνου, όπως σημειώθηκε, είναι καθυστέρηση στην επικοινωνία του κόμβου με τον δαίμονα. Εντούτοις, το αποτέλεσμα αυτό είναι θετικό για τις δυνατότητες κλιμάκωσης του συστήματος. Παρά το γεγονός πως η αντιστοίχιση καθυστερεί περισσότερο, παρατηρούμε τον συνολικό χρόνο που δαπανήθηκε για τη λειτουργία να είναι αρχικά περισσότερος από ότι θα ήταν με έναν πελάτη, όπως αναμένεται, όμως καθώς αυξάνουμε τα σύνολα αρχείων, η καθυστέρηση μειώνεται αισθητά, και καταλήγει να έχει προβάδισμα αυτή η εκτέλεση. Η απόδοση αυτή οφείλεται αποκλειστικά στο πρωτόκολλο επικοινωνίας, και μπορούμε μόνο να υποθέσουμε ότι η καθυστέρηση που δημιουργείται μεταξύ του πελάτη και του κόμβου NSDB μειώνει τον φόρτο του εξυπηρετητή αρχείων, με αποτέλεσμα να αποστέλλονται τα δεδομένα γρηγορότερα. Η προηγούμενη υπόθεση για το NFSv4 επαληθεύεται στο σχήμα 5.2.5, όπου ο χρόνος αντιστοίχισης είναι ανάλογος με αυτόν που απαιτεί ένας πελάτης για να εκτελέσει την αντιστοίχιση τοπικά, ενώ ο συνολικός χρόνος

που απαιτείται είναι διπλάσιος από αυτόν που βλέπουμε στο σχήμα 5.2.4, και σαφώς υψηλότερος από αυτόν που απαιτείται για κάθε πελάτη ξεχωριστά, όπως φαίνεται στο σχήμα 5.2.1. Αυτό μας οδηγεί τελικά να συμπεράνουμε, πως όταν ο φόρτος του εξυπηρετητή αρχείων είναι υψηλός, τότε η απόδοση του NFSv4 χωλαίνει, με αποτέλεσμα αφενός η αντιστοίχιση στον εξυπηρετητή να μειώνει σημαντικά τον χρόνο αντιστοίχισης, και αφετέρου να επιβαρύνει τον κόμβο, δημιουργώντας καθυστέρηση στην εξυπηρέτηση του πελάτη.

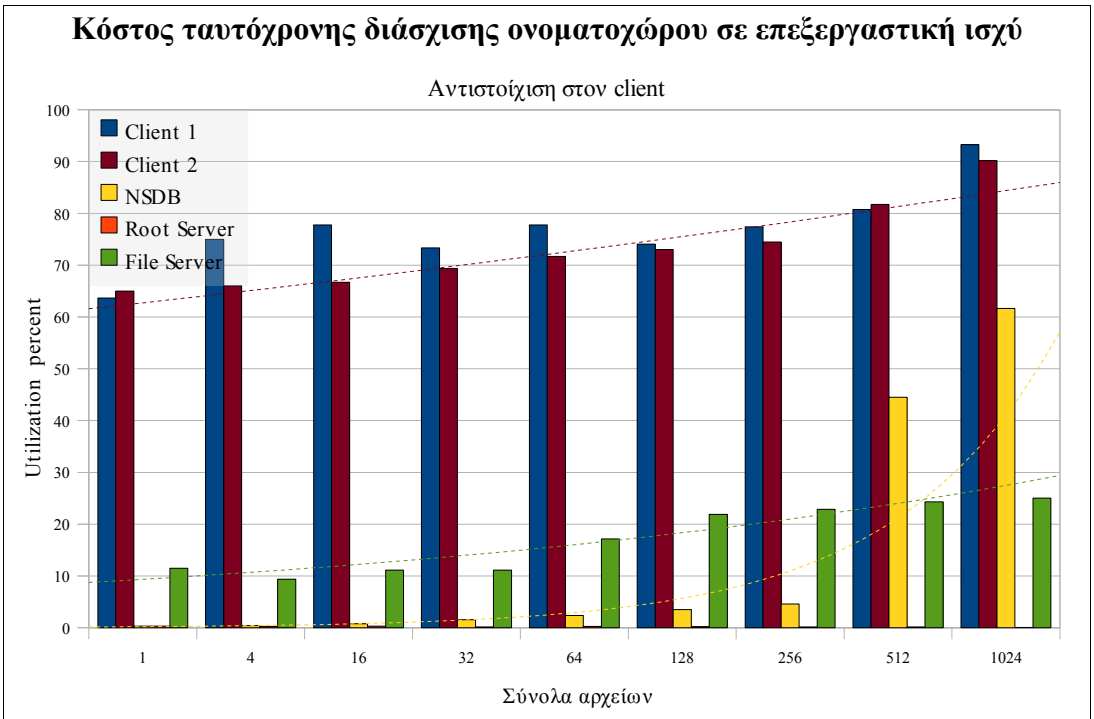


Σχήμα 5.2.6: Σύγκριση κόστους ταυτόχρονης διάσχισης δένδρου καταλόγων του kernel 2.6.29.4 σε εύρος ζώνης δικτύου από δύο πελάτες, με την αντιστοίχιση να γίνεται στον client και στον server. Παρατηρούμε και πάλι, ότι τα ανταλλασσόμενα δεδομένα αυξάνονται με τον αριθμό των συνόλων αρχείων, ενώ η αντιστοίχιση δεν επιβαρύνει ιδιαίτερα τον κόμβο NSDB, αν παρατηρήσουμε την καθυστέρηση που δέχονται ο εξυπηρετητής αρχείων και οι πελάτες από τη διάσχιση.

Κόστος ταυτόχρονης διάσχισης. Στο σχήμα 5.2.6 φαίνεται ο όγκος δεδομένων που ελήφθησαν κατά τη διάρκεια των πειραμάτων από τους κόμβους του συστήματος. Είναι εμφανές, όπως και προηγουμένως, ότι ο κόμβος NSDB και ο εξυπηρετητής ρίζας δεν ανταλλάσσουν σημαντικό μέγεθος δεδομένων, παρόλο που για τον πρώτο το μέγεθος αυξάνεται, όπως αναμένεται, με την αύξηση του αριθμού των αιτήσεων. Η τάξη μεγέθους αυτή αποτελεί και τον όγκο των δεδομένων που παράγεται λόγω των αιτήσεων αντιστοίχισης, ενώ ανάλογος είναι και ο όγκος δεδομένων με τον οποίο επιβαρύνεται ο κόμβος που την αναλαμβάνει. Στους πελάτες, τα δεδομένα που ανταλλάσσονται δεν διαφέρουν και πολύ στις δύο περιπτώσεις, καθώς ο αριθμός αιτήσεων είναι ο ίδιος, και το μικρό μέγεθος δεδομένων που τις εκφράζει χάνεται στην ποσότητα των δεδομένων που απαιτούνται για την εκτέλεση της λειτουργίας για μικρό αριθμό συνόλων αρχείων, όμως για μεγαλύτερους αριθμούς παρατηρούμε ότι τα δεδομένα αντιστοίχισης είναι εμφανή στον υπεύθυνο κόμβο. Επίσης, τα δεδομένα που αποστέλλονται από τον εξυπηρετητή αρχείων είναι διπλάσια από αυτά του σχήματος 5.2.2, που είναι αναμενόμενο, καθώς τώρα εξυπηρετούνται δύο πελάτες.

Από πλευράς επεξεργαστικού φόρτου, υπάρχει διαφορά και πάλι κυρίως στην περίπτωση του πελάτη, καθώς παρατηρούμε ότι τα δεδομένα φτάνουν αρκετά γρήγορα, ώστε να

παρατηρείται συνεχής αύξηση της επεξεργαστικής ισχύος που χρησιμοποιείται. Μέσω του σχήματος αυτού, μάλιστα, γίνεται αντιληπτό και αυτό που ειπώθηκε προηγουμένως. Όπως φαίνεται και από το σχήμα 5.2.7, ο φόρτος του εξυπηρετητή αρχείων είναι μικρότερος, ακόμα και στο ήμισυ, από τα αποτελέσματα που παρατηρήσαμε στο σχήμα 5.2.3, ενώ αντίθετα, ο φόρτος του κόμβου NSDB φαίνεται να έχει αυξηθεί, με μεγαλύτερη αύξηση να παρατηρείται στις δύο τελευταίες περιπτώσεις, όπου έχουμε τον μικρότερο χρόνο εξυπηρέτησης.



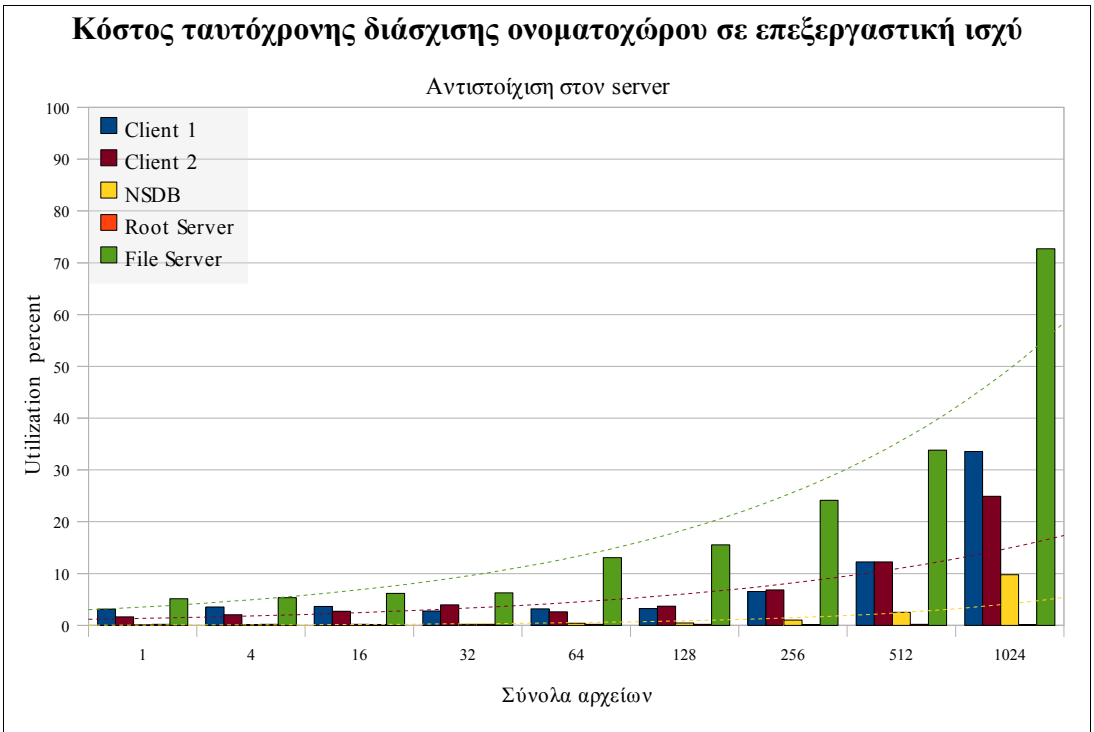
Σχήμα 5.2.7: Μεταβολή επεξεργαστικού κόστους διάσχισης δένδρου καταλόγων του kernel 2.6.29.4 κατά την αύξηση του αριθμού συνόλων αρχείων, με την αντιστοίχιση να γίνεται στον client. Παρατηρούμε ότι στους πελάτες παρατηρείται αύξηση του επεξεργαστικού φόρτου, η οποία οφείλεται στον μειωμένο φόρτο του εξυπηρετητή αρχείων που τώρα μπορεί να τους τροφοδοτεί με περισσότερα δεδομένα. Όπως αναμένεται, ο κόμβος NSDB έχει αυξημένο φόρτο σε σχέση με το προηγούμενο πείραμα, στις περιπτώσεις πολλών αιτήσεων.

Τέλος, στο σχήμα 5.2.8 φαίνεται ο επεξεργαστικός φόρτος στην περίπτωση αντιστοίχισης στον εξυπηρετητή. Παρατηρούμε αυτό που είδαμε και στο προηγούμενο πείραμα, δηλαδή ο πελάτης χρησιμοποιεί ελάχιστα τον επεξεργαστή, αναμένοντας τον περισσότερο χρόνο για τα δεδομένα. Επιπλέον, το γεγονός πως οι αιτήσεις γίνονται μονάχα από τον εξυπηρετητή, και μια φορά για όλους τους πελάτες, μειώνει αισθητά τον επεξεργαστικό φόρτο του κόμβου NSDB, όμως αυξάνει αρκετά τον φόρτο του εξυπηρετητή αρχείων.

5.2.3 Συμπέρασμα πειράματος

Όπως έγινε αντιληπτό, είναι δύσκολο να επιλεγεί μια από τις δύο υλοποιήσεις ως η καλύτερη για το πρόβλημα της διάσχισης. Αφενός, όταν η αντιστοίχιση γίνεται στον πελάτη, αποφεύγουμε τον φόρτο που δημιουργείται στον εξυπηρετητή αρχείων, με αποτέλεσμα να επιτελείται ταχύτερα η εξυπηρέτηση του πρώτου. Όμως, αυτό έχει κόστος στον κόμβο

NSDB, ο οποίος αναλαμβάνει έτσι να εξυπηρετεί τον διπλάσιο αριθμό αιτήσεων, με αντίκτυπο στα δεδομένα που μεταφέρονται μέσω δικτύου, και στον επεξεργαστικό φόρτο που δέχεται. Από την άλλη, αν η αντιστοίχιση γίνει στον εξυπηρετητή, τότε τα σχετικά δεδομένα πρέπει να μεταφερθούν μαζί με τα υπόλοιπα, επιβαρύνοντας το δίκτυο και τον εξυπηρετητή αρχείων.



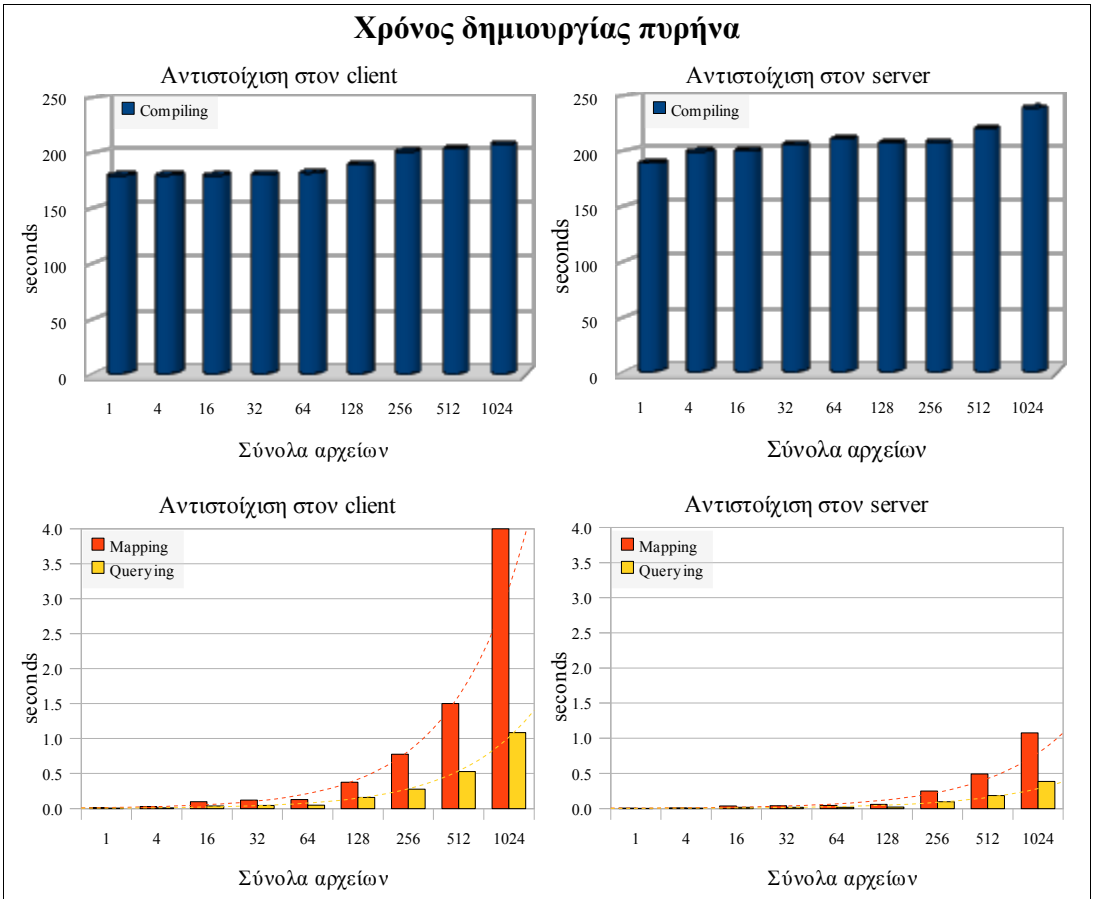
Σχήμα 5.2.8: Μεταβολή επεξεργαστικού κόστους διάσχισης δένδρου καταλόγων του kernel 2.6.29.4 κατά την αύξηση του αριθμού συνόλων αρχείων, με την αντιστοίχιση να γίνεται στον server. Παρατηρούμε ότι ο φόρτος του εξυπηρετητή που έχει αναλάβει και την αντιστοίχιση είναι έντονος, αντίθετα με τον κόμβο NSDB, ο οποίος εξυπηρετεί τις μισές αιτήσεις και μένει περισσότερο αδρανής.

5.3 Κόστος κατανεμημένης δημιουργίας λογισμικού

Σκοπός πειράματος. Το πείραμα αυτό έχει ως σκοπό την αξιολόγηση της επιβάρυνσης που δέχεται το σύστημα, όταν ένας χρήστης εκτελέσει κάποια λειτουργία που μεταχειρίζεται ομοσπονδιακά δεδομένα. Ο σκοπός του πειράματος είναι να προσομοιώσει ένα απαιτητικό και κατά το δυνατόν ρεαλιστικό σενάριο, στο οποίο η λειτουργία δεν επεξεργάζεται μόνο μεταδεδομένα, αλλά και πραγματικά δεδομένα από αρχεία του ονοματοχώρου. Έτσι, μπορεί να εκτιμηθεί κατά πόσο μπορούμε να θεωρήσουμε την ομοσπονδία αξιόπιστη για πραγματική χρήση, παρά την καθυστέρηση που εισάγεται από τον μηχανισμό αντιστοίχισης.

Περιγραφή πειράματος. Το δένδρο καταλόγων που χρησιμοποιούμε ως ονοματοχώρο είναι και πάλι αυτό που προκύπτει από τον πηγαίο κώδικα του πυρήνα 2.6.29.4 [Kernel29.4], ενώ η εφαρμογή που χρησιμοποιούμε είναι η δημιουργία πυρήνα, το αρχείο ρυθμίσεων του

οποίου δημιουργείται από το πρότυπο *allnoconfig*[3]. Η επιλογή της εν λόγω ρύθμισης θεωρείται ιδανική, καθώς επιλέγει σχεδόν ένα χαρακτηριστικό από κάθε φάκελο του ονοματοχώρου, και έτσι δεν αναλώνεται μεγάλο χρονικό διάστημα σε κάποιο σημείο του. Ο λόγος που αυτό είναι σημαντικό, μπορεί να φανεί από τα προηγούμενα πειράματα. Ο χρόνος δημιουργίας ενός kernel μπορεί να κρατήσει έως και δεκάδες λεπτά ενώ η διάσχιση ολόκληρου του ονοματοχώρου επιβαρύνει το σύστημα, στην χειρότερη περίπτωση με μερικά δευτερόλεπτα. Έτσι, ο μόνος τρόπος να έχουμε αποτελέσματα που εστιάζουν στο ζητούμενο, είναι να εκτελεστεί μόνο μια λειτουργία ανάγνωσης/εγγραφής σε κάθε φάκελο, η οποία κατά προτίμηση να μην δαπανά πολύ χρόνο. Βέβαια, ακόμα και με την ρύθμιση *allnoconfig*, ο χρόνος δημιουργίας δεν είναι συγκρίσιμος της προστιθέμενης καθυστέρησης, όμως είναι ο μοναδικός κανονικοποιημένος τρόπος που μας επιτρέπει να εξάγουμε κάποια συμπεράσματα. Για την οργάνωση των πειραμάτων, ακολουθείται η στρατηγική μετρήσεων που περιγράφηκε και προηγουμένως.

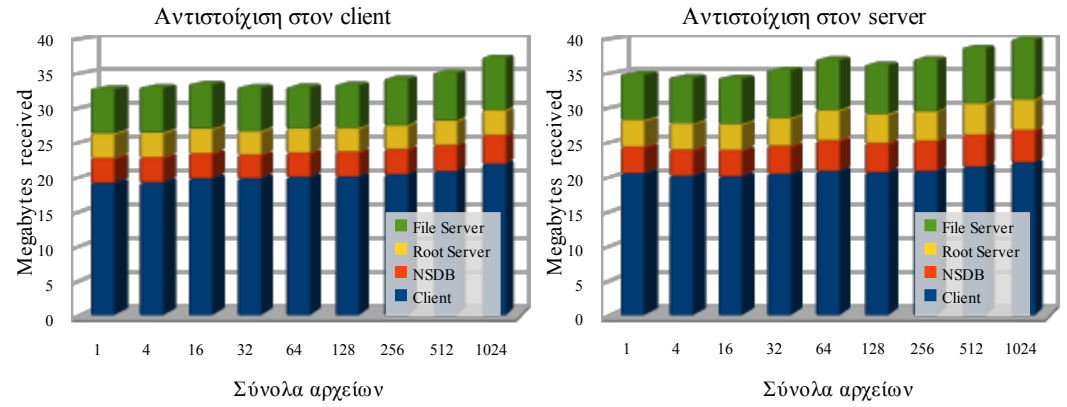


Σχήμα 5.3.1: Σύγκριση χρόνου δημιουργίας του kernel 2.6.29.4, με την αντιστοίχιση να γίνεται στον client και στον server. Παρατηρούμε ότι η αύξηση του χρόνου εξαρτάται από τον αριθμό συνόλων αρχείων, ενώ η αντιστοίχιση καταλαμβάνει 0.01-1.93% στον πελάτη και 0.003-0.449% στον εξυπηρετητή. Επίσης, ο φόρτος του πελάτη φαίνεται να καθυστερεί σημαντικά την επίλυση των αντιστοιχίσεων, όταν τις αναλαμβάνει ο ίδιος.

[3]Το αρχείο ρυθμίσεων που προκύπτει από το πρότυπο allnoconfig επιλέγει ελάχιστα χαρακτηριστικά για τον νέο πυρήνα, και δημιουργείται εισάγοντας την εντολή: `make allnoconfig ARCH=`"*arch*", όπου "*arch*" είναι η αρχιτεκτονική του υπολογιστή για τον οποίο προορίζεται ο πυρήνας (εδώ χρησιμοποιήθηκε η αρχιτεκτονική `x86`).

5.3.1 Δημιουργία πυρήνα από έναν πελάτη

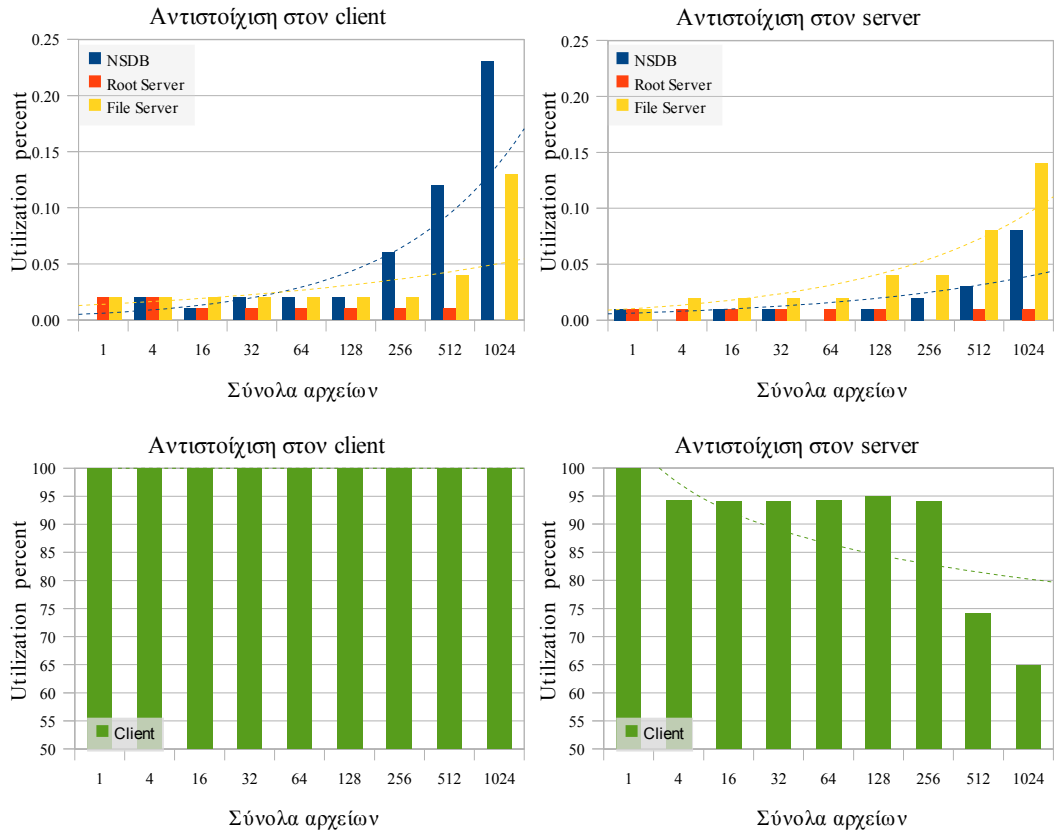
Χρόνος δημιουργίας πυρήνα. Ο χρόνος που χρειάστηκε για το πείραμα φαίνεται στο σχήμα 5.3.1, όπου η αναμενόμενη διαφορά μεταξύ του χρόνου δημιουργίας και της καθυστέρησης που προκαλεί η αντιστοίχιση είναι εμφανής. Εδώ όμως παρατηρούμε, πως το χάσμα μεταξύ του χρόνου που απαιτείται για το χρόνο αντιστοίχισης στον πελάτη και στον εξυπηρετητή είναι μεγάλο. Αυτό οφείλεται αποκλειστικά στο γεγονός πως η δημιουργία του πυρήνα καταλαμβάνει σχεδόν ολόκληρη την επεξεργαστική ισχύ του συστήματος του πελάτη, με αποτέλεσμα την έντονη καθυστέρηση στην αποστολή και την επεξεργασία των μηνυμάτων που ανταλλάσσονται με τον κόμβο NSDB. Η υπεροχή της δεύτερης υλοποίησης όμως δεν ισχύει και για τον συνολικό χρόνο που απαιτείται για την λειτουργία, καθώς τότε τα δεδομένα της αντιστοίχισης πρέπει να σταλούν από το ήδη βεβαρημένο δίκτυο, με αποτέλεσμα καθυστέρηση που μπορεί να φτάσει ακόμα και το 15% του απαιτούμενου χρόνου για τη λειτουργία. Αυτό το ποσοστό προφανώς μειώνεται όσο αυξάνεται ο εξ' ορισμού χρόνος που απαιτεί η λειτουργία, όμως παρόλα αυτά παραμένει ικανό να εξισορροπήσει τα δύο σενάρια.



Σχήμα 5.3.2: Σύγκριση κόστους δημιουργίας του kernel 2.6.29.4 σε εύρος ζώνης, με την αντιστοίχιση να γίνεται στον client και στον server. Παρατηρούμε ότι ο όγκος των δεδομένων που απαιτείται είναι τέτοιος, που δεν επιτρέπει να διακρίνουμε καθαρά το τμήμα που αφορά την αντιστοίχιση. Αξιοσημείωτο είναι το φαινόμενο της παλαίωσης των δεδομένων αντιστοίχισης λόγω μεγάλης διάρκειας της λειτουργίας, το οποίο οδηγεί στην επανάληψη τους, αυξάνοντας τον φόρτο του NSDB και του ριζικού εξυπηρετητή.

Κόστος δημιουργίας πυρήνα. Στο σχήμα 5.3.2 αναλύεται το κόστος του πειράματος σε εύρος ζώνης δικτύου. Όπως φαίνεται από τα σχεδιαγράμματα, ο όγκος των δεδομένων που ανταλλάσσονται είναι σαφώς μεγαλύτερος από αυτόν του προηγούμενου πειράματος, καθώς τώρα λαμβάνονται ολόκληρα αρχεία πηγαίου κώδικα, από τα οποία σχηματίζονται τοπικά τα αντίστοιχα εκτελέσιμα. Παρατηρούμε ότι τα δεδομένα που λαμβάνονται είναι κατά πολύ περισσότερα στον πελάτη από ότι στον εξυπηρετητή αρχείων, αλλά αυτό είναι αναμενόμενο αφού εκεί γίνεται η επεξεργασία τους. Αυτό κάνει πιο δύσκολο να αντιληφθούμε τον όγκο

των δεδομένων που επιβαρύνουν τον κόμβο που κάνει την αντιστοίχιση, παρόλο η τελευταία επανάληψη μπορεί να βοηθήσει σε αυτό. Αξιοσημείωτος είναι ο όγκος των δεδομένων στον ριζικό εξυπηρετητή και στον κόμβο NSDB, αν σκεφτούμε ότι οι αιτήσεις που εκτελούνται είναι ακριβώς οι ίδιες με το πείραμα της ενότητας 5.2. Ο λόγος που παρατηρείται αυτό το φαινόμενο οφείλεται στο γεγονός πως οι πληροφορίες που αφορούν τις αντιστοιχίσεις έχουν χρόνο λήξης, μετά τον οποίο πρέπει να ανανεωθούν για να αποφευχθεί πιθανή αλλαγή στη θέση των δεδομένων, χωρίς να γίνει αυτό αντιληπτό από τον πελάτη. Λόγω της μεγάλης διάρκειας του πειράματος, η αντιστοίχιση που δείχνει στη ρίζα ανανεώνεται αρκετές φορές με αποτέλεσμα να προκύπτει ο φόρτος του εξυπηρετητή ρίζας. Για τον κόμβο NSDB τα πράγματα είναι περισσότερο πολύπλοκα. Ο όγκος δεδομένων που παρατηρούμε δεν οφείλεται στην επανάληψη όλων των αιτήσεων, καθώς σε καταλόγους μεγάλου βάθους, γίνονται ελάχιστες εργασίες. Οι περισσότερες επαναλήψεις γίνονται κοντά στην ρίζα, και ειδικότερα στους φακέλους που περιέχουν πολλούς υποφακέλους. Αυτό ενισχύει την εξήγηση που δόθηκε για τον αυξημένο φόρτο του ριζικού εξυπηρετητή.



Κόστος δημιουργίας πυρήνα σε επεξεργαστική ισχύ

Σχήμα 5.3.3: Σύγκριση κόστους δημιουργίας του kernel 2.6.29.4 σε επεξεργαστική ισχύ, με την αντιστοίχιση να γίνεται στον client και στον server. Ο αυξημένος επεξεργαστικός φόρτος του πελάτη στην πρώτη περίπτωση δεν οφείλεται αποκλειστικά στην λειτουργία, καθώς ο κόμβος επιβαρύνεται από τις αιτήσεις αντιστοίχισης, με αποτέλεσμα παρόλο που ο επεξεργαστής λειτουργεί στο μέγιστο των δυνατοτήτων του, ο απαιτούμενος χρόνος να αυξάνεται. Στη δεύτερη περίπτωση ο χρόνος που αναλώνεται για την παραλαβή των δεδομένων αυξάνεται σταθερά, και έτσι δεν υφίστανται αρκετά για την πλήρη εκμετάλλευση του επεξεργαστή.

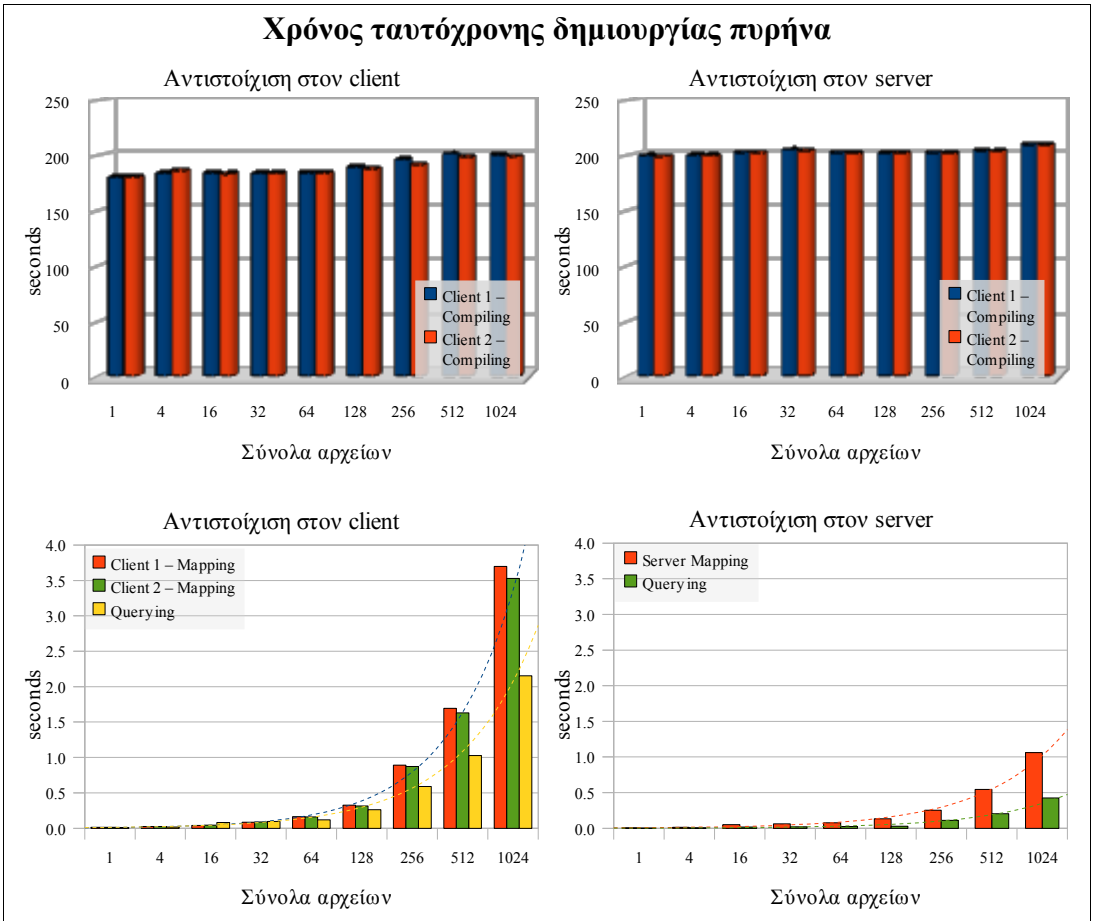
Όσων αφορά την επεξεργαστική ισχύ που χρησιμοποιείται, τα αποτελέσματα συμφωνούν και με το προηγούμενο πείραμα. Όταν η αντιστοίχιση γίνεται στον πελάτη, παρατηρείται έντονος φόρτος στον κόμβο NSDB, ο οποίος, όμως, είναι αισθητά μεγαλύτερος από αυτόν του σχήματος 5.2.3, λόγω του φαινομένου επανάληψης των αιτήσεων. Εντούτοις, όταν η αντιστοίχιση λαμβάνει χώρα στον εξυπηρετητή, ο φόρτος του NSDB μειώνεται, ενώ υπάρχει μεγαλύτερη επιβάρυνση στον ίδιο. Τέλος, ο επεξεργαστικός φόρτος του πελάτη στην πρώτη περίπτωση είναι έντονος, καθώς ο επεξεργαστής λειτουργεί στο μέγιστο των δυνατοτήτων του. Όμως, από τα προηγούμενα δεδομένα μπορούμε να συμπεράνουμε, πως όσο ο αριθμός των συνόλων δεδομένων αυξάνεται τόσο μικρότερο μέρος του φόρτου χρησιμοποιείται για τη λειτουργία, ενώ το υπόλοιπο εκχωρείται για την αντιστοίχιση δίνοντας τελικά την αύξηση που παρατηρούμε στον συνολικό χρόνο. Στην δεύτερη περίπτωση όπου τα μεταδεδομένα αποστέλλονται από τον εξυπηρετητή αρχείων, παρατηρούμε ότι δημιουργείται καθυστέρηση, καθώς υπάρχει αρκετή πληροφορία για να εκμεταλλευτεί τις δυνατότητες του επεξεργαστή μόνο στις πρώτες επαναλήψεις. Η μεγάλη καθυστέρηση που αυτό συνεπάγεται, μας οδηγεί να αναλογιστούμε την αντιστοίχιση στον εξυπηρετητή ως ιδανική για απαιτητικές λειτουργίες.

5.3.2 Ταυτόχρονη δημιουργία πυρήνα από δύο πελάτες

Όπως και προηγουμένως, στην δεύτερη εκδοχή του πειράματος χρησιμοποιούμε δύο πελάτες, οι οποίοι ταυτόχρονα δημιουργούν τον ίδιο πυρήνα από τα αρχεία πηγαίου κώδικα που διαθέτει η ομοσπονδία. Τα εκτελέσιμα αρχεία δημιουργούνται τοπικά για να αποφευχθεί εμπλοκή μεταξύ των δύο πελατών. Αυτό γίνεται με έναν μηχανισμό που πριν το πείραμα διατρέχει ολόκληρο τον ονοματοχώρο, και δημιουργεί ένα αντίγραφό του τοπικά, ώστε κάθε τοπικό αρχείο να αποτελεί σύνδεσμο προς το αντίστοιχο αρχείο στον ονοματοχώρο της ομοσπονδίας. Έτσι, τα εκτελέσιμα παράγονται τοπικά, ενώ κάθε φορά που ο πελάτης προσπαθεί να προσπελάσει ένα αρχείο, γίνονται όλες οι αντιστοιχίσεις που απαιτούνται για να βρεθεί το αρχείο στον ονοματοχώρο, διασχίζοντας όλους τους φακέλους της ομοσπονδίας που περιλαμβάνονται στο μονοπάτι του.

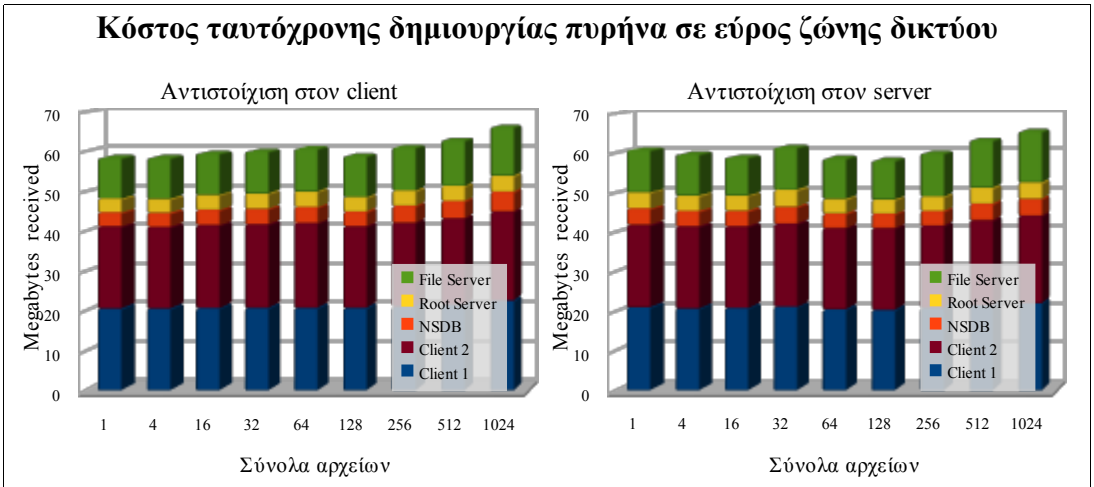
Χρόνος ταυτόχρονης δημιουργίας πυρήνων. Όπως φαίνεται στο σχήμα 5.3.4, ο χρόνος που απαιτείται για την ταυτόχρονη δημιουργία πυρήνα από δύο πελάτες προσεγγίζει αυτόν που είδαμε στο σχήμα 5.3.1, με τη διαφορά πως παρατηρείται κατά μέσο όρο μικρή αύξηση στον συνολικό χρόνο της λειτουργίας. Η μικρή αυτή αύξηση οφείλεται στην καθυστέρηση του κόμβου NSDB να εξυπηρετήσει τις αιτήσεις που δέχεται ταυτόχρονα και από τους δύο πελάτες. Το γεγονός πως τα δεδομένα χρειάζονται περισσότερο χρόνο για την επεξεργασία από ότι για τη λήψη τους ενισχύει αυτό το επιχείρημα. Ένα ακόμα στοιχείο αποτελεί ο χρόνος αντιστοίχισης που φαίνεται στο αντίστοιχο σχήμα. Εκεί παρατηρούμε πως στις περιπτώσεις

που ο χρόνος αυτός ξεπερνά τον αντίστοιχο χρόνο του σχήματος 5.3.1, παρατηρείται αύξηση του συνολικού χρόνου, ενώ όταν είναι μικρότερος, παρατηρείται μείωση. Παρόλα αυτά, όμως, είναι εμφανές πως πάλι οι δύο πελάτες χρειάζονται το ίδιο χρονικό διάστημα για να ολοκληρώσουν τη λειτουργία και να επιλύσουν τις αντιστοιχίσεις που χρειάζονται.

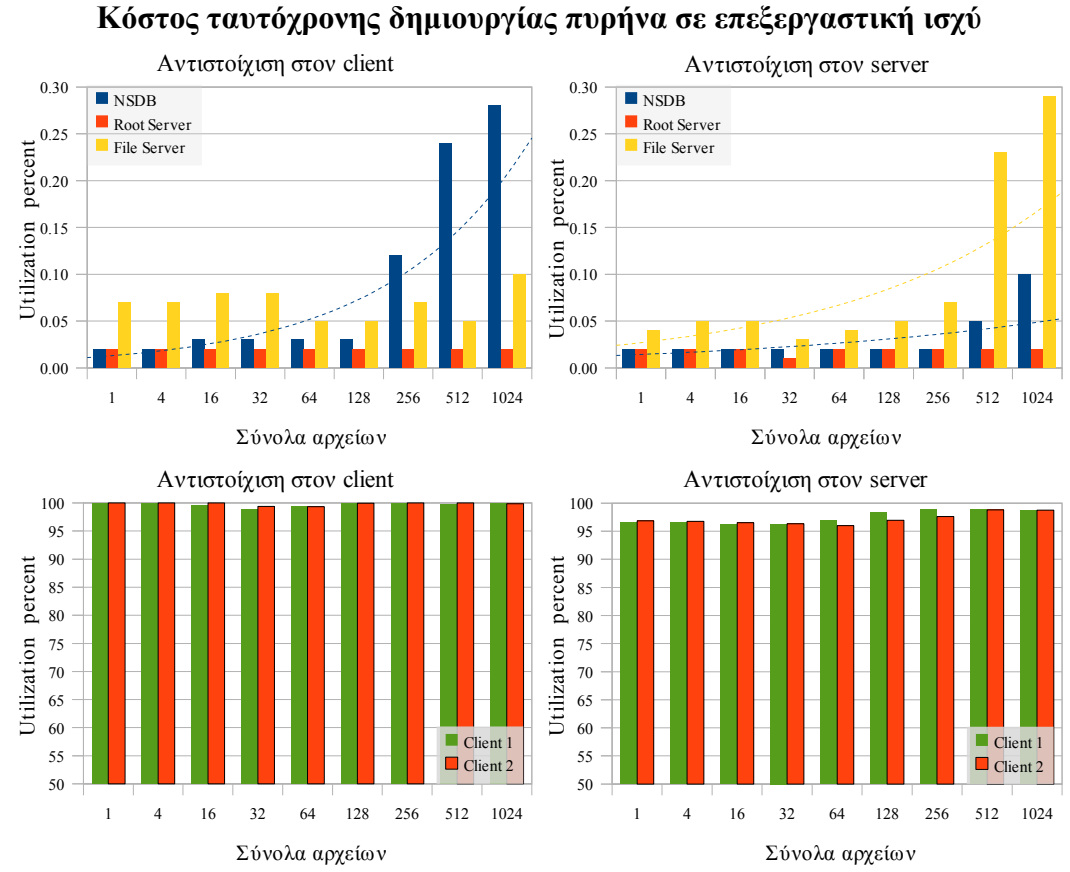


Σχήμα 5.3.4: Σύγκριση χρόνου δημιουργίας του kernel 2.6.29.4, με την αντιστοίχιση να γίνεται στον client και στον server. Παρατηρούμε ότι η αύξηση του χρόνου αντιστοίχισης είναι ανάλογη αυτής των συνόλων αρχείων, ενώ και πάλι η αντιστοίχιση στον πελάτη φαίνεται να καθυστερεί σημαντικά λόγω του επεξεργαστικού φόρτου.

Κόστος ταυτόχρονης δημιουργίας πυρήνων. Ο φόρτος που δημιουργείται στο δίκτυο κατά την παράλληλη εκτέλεση του πειράματος είναι αναμενόμενος, καθώς, όπως φαίνεται στο σχήμα 5.3.5, αυξάνεται λόγω του δεύτερου πελάτη. Ο φόρτος του ριζικού εξυπηρετητή και του κόμβου NSDB επεξηγήθηκε και προηγουμένως, ενώ πρατηρούμε ότι στους δύο πελάτες ο φόρτος είναι παρόμοιος. Παρόλα αυτά, ο όγκος των συνολικών δεδομένων δεν μας βοηθά να παρατηρήσουμε τον φόρτο οφείλεται στην επίλυση των αντιστοιχίσεων για τους κόμβους που τις αναλαμβάνουν. Αυτό συμβαίνει επειδή η τάξη μεγέθους έχει σχεδόν διπλασιαστεί, με αποτέλεσμα η διαφορά, που από τις μετρήσεις κυμαίνεται από 200 έως 300 Kilobytes, να χάνεται στο σύνολο των Megabytes που λαμβάνονται από κάθε κόμβο. Το διάστημα αυτό προέκυψε για όλες τις επαναλήψεις του πειράματος από τα αριθμητικά αποτελέσματα.



Σχήμα 5.3.5: Σύγκριση κόστους δημιουργίας του kernel 2.6.29.4 σε εύρος ζώνης δικτύου, με την αντιστοίχιση να γίνεται στον client και στον server. Παρατηρούμε ότι η αύξηση των ανταλλασσώμενων δεδομένων είναι ανάλογη αυτής των συνόλων αρχείων, ενώ η αντιστοίχιση δημιουργεί επιβάρυνση κυρίως στον εξυπηρετητή αρχείων και στον πελάτη. Ο συνολικός όγκος των δεδομένων δεν μας επιτρέπει να διακρίνουμε τα δεδομένα που οφείλονται στην αντιστοίχιση, καθώς αποτελούν το 0.9% στον πελάτη, και το 1.9% στον εξυπηρετητή.



Σχήμα 5.3.6: Σύγκριση κόστους δημιουργίας του kernel 2.6.29.4 σε επεξεργαστική ισχύ, με την αντιστοίχιση να γίνεται στον client και στον server. Ο επεξεργαστής των πελατών χρησιμοποιείται πλήρως και στις δύο περιπτώσεις, αλλά και στις δύο η πραγματική επεξεργαστική ισχύς που χρησιμοποιείται για την λειτουργία μειώνεται, και ένα μέρος της καταλαμβάνεται από το κόστος της αντιστοίχισης.

Στο επίπεδο της απαιτούμενης επεξεργαστικής ισχύος, τα αποτελέσματα του σχήματος 5.3.6 φαίνεται αρχικά να δείχνουν πως και στις δύο περιπτώσεις, ο επεξεργαστής του πελάτη

χρησιμοποιείται στο μέγιστο των δυνατοτήτων του για την εκτέλεση της λειτουργίας, όμως αυτό δεν συμβαίνει στην πραγματικότητα. Αν λάβουμε υπόψιν την αύξηση που παρατηρείται στη συνολική διάρκεια της δημιουργίας του πυρήνα, μπορούμε να συμπεράνουμε πως ο πραγματικός επεξεργαστικός φόρτος που αφιερώνεται στην λειτουργία μειώνεται, ενώ την θέση του καταλαμβάνει ο απαιτούμενος φόρτος της αντιστοίχισης, ο οποίος φαίνεται ότι δεν είναι απαγορευτικός, καθώς οι καθυστερήσεις μεταξύ των επαναλήψεων είναι μικρές. Στην δεύτερη περίπτωση, όμως, παρά την αύξηση των συνόλων αρχείων, παρατηρούμε ότι υπάρχει χρονικό διάστημα, που ο επεξεργαστής μένει αδρανής. Αν συμβουλευτούμε το σχήμα 5.3.3, βλέπουμε πως σε κανονικές συνθήκες ο φόρτος θα έπρεπε να μειώνεται, ίσως και στους δύο πελάτες, με τον ίδιο ρυθμό. Εδώ όμως, το γεγονός πως ο εξυπηρετητής αναλαμβάνει την αντιστοίχιση βοηθά, καθώς οι αιτήσεις που αποστέλλονται από τον δεύτερο πελάτη για τα δεδομένα που έχει ζητήσει ο πρώτος[4], ανανεώνουν και τον χρόνο που έχει παρέλθει από την αναζήτηση της εκάστοτε αντιστοίχισης, καλύπτοντας σε μεγάλο ποσοστό την ανάγκη για ανανέωση τους τη στιγμή που θα τα χρειαστεί ξανά κάποιος από τους δύο πελάτες. Αυτό το φαινόμενο μειώνει τον χρόνο αναμονής του πελάτη, αυξάνοντας τον μέσο επεξεργαστικό φόρτο που φαίνεται στο σχήμα. Ο κόμβος NSDB είναι και πάλι περισσότερο βεβαρημένος στην πρώτη περίπτωση, ενώ ο φόρτος μεταφέρεται στον εξυπηρετητή αρχείων, όταν η αντιστοίχιση λαμβάνει χώρα εκεί. Αν συγκρίνουμε αυτά τα αποτελέσματα με τα αντίστοιχα του σχήματος 5.3.3, παρατηρούμε πως ο φόρτος του κόμβου NSDB κινείται στα ίδια επίπεδα, ενώ αυτός του εξυπηρετητή αρχείων έχει αυξηθεί. Το πρώτο αποτέλεσμα οφείλεται στο γεγονός ότι οι αιτήσεις είναι ούτως ή άλλως αραιές σε σχέση με το χρόνο του πειράματος, επομένως εξυπηρετούνται το ίδιο γρήγορα. Ο μεγαλύτερος φόρτος στον εξυπηρετητή αρχείων είναι και αυτός προβλέψιμος, καθώς προέρχεται από την αύξηση της ζήτησης στα δεδομένα με την προσθήκη του δεύτερου πελάτη.

5.3.3 Συμπέρασμα πειράματος

Από τα προηγούμενα, έγινε σαφής ο λόγος εκτέλεσης του συγκεκριμένου πειράματος, σε αντιπαράθεση με την απλή διάσχιση του ονοματοχώρου. Στην περίπτωση τόσο απαιτητικών λειτουργιών βλέπουμε ότι το βάρος για την βελτίωση της απόδοσης πέφτει σε πολύ μεγάλο βαθμό στο πρωτόκολλο επικοινωνίας, καθώς το μεγαλύτερο μέρος του χρόνου καθυστέρησης στην εκτέλεση της λειτουργίας οφείλεται στην μεταφορά των δεδομένων μέσω δικτύου, καθώς και στην ανανέωση της πληροφορίας αντιστοίχισης. Η τελευταία σχετίζεται και με το πρωτόκολλο επικοινωνίας, καθώς μετά την παραλαβή της πληροφορίας από τον κόμβο NSDB, που διαρκεί από 0.7 έως 1.5 milliseconds, πρέπει να ανανεωθεί η αναφορική σύνδεση (*referral*), που δημιουργείται από το NFSv4 για την απόκτηση της πληροφορίας.

[4]Ως "πρώτο", θεωρούμε τον πρώτο πελάτη που εξυπηρετήθηκε, και όχι κατ' ανάγκη τον πελάτη 1 του σχήματος.

Όσων αφορά την σύγκριση μεταξύ των δύο υλοποιήσεων, παρατηρούμε πως το τοπίο και πάλι είναι θολό, καθώς και στις δύο περιπτώσεις έχουμε θετικά και αρνητικά στοιχεία, που αφορούν κυρίως το σύστημα στο οποίο πρόκειται να χρησιμοποιηθούν. Όμως, επειδή το είδος της λειτουργίας αυτής διασπείρει τις αιτήσεις αντιστοίχισης στο χρόνο εκτέλεσης, ο φόρτος του κόμβου NSDB δεν είναι έντονος, με αποτέλεσμα να τις εξυπηρετεί γρήγορα, κάνοντας την αντιστοίχιση στον πελάτη ελκυστική. Από την άλλη, αν οι πελάτες είναι πολλοί και ζητούν πρόσβαση σε παρόμοιους φακέλους, τότε είδαμε πως όταν ο εξυπηρετητής εκτελεί την αντιστοίχιση παίρνουμε πιο μικρούς χρόνους, κάτι που θα γινόταν ιδιαίτερα εμφανές σε ένα περιβάλλον με εκατοντάδες πελάτες, οι οποίοι θα ήταν ικανοί να αυξήσουν τον επεξεργαστικό φόρτο του NSDB αρκετά αν αναλάμβαναν οι ίδιοι την αντιστοίχιση.

6 *Σχετικές Εργασίες*

"σχετικός", -η, -ό επίθ. *που έχει σχέση, αναφορά, σύνδεση με κάτι*

Στην συνέχεια θα προσπαθήσουμε να αναλύσουμε τις σημαντικότερες από τις υπάρχουσες υλοποιήσεις που δύνανται να δημιουργήσουν ένα κατανεμημένο περιβάλλον σε επίπεδο συστήματος αρχείων, και τους λόγους για τους οποίους αποτυγχάνουν να το πετύχουν σε ομοσπονδιακό επίπεδο, σε συνάρτηση με τις απαιτήσεις που τέθηκαν.

6.1 Common Internet Filesystem

Το πρωτόκολλο CIFS χρησιμοποιείται για την αίτηση υπηρεσιών πάνω σε αρχεία και άλλους διαθέσιμους πόρους ενός δικτύου. Βασίζεται στο πρωτόκολλο SMB της Microsoft, και είναι ανοιχτού κώδικα. Σημαντικά χαρακτηριστικά που αφορούν άμεσα την δημιουργία κοινού ονοματοχώρου που πληροί τις απαιτήσεις ομοσπονδιακών συστημάτων αρχείων είναι οι δυνατότητες κλειδώματος αρχείων για προσπέλαση από πελάτες, προσωρινής τοπικής αποθήκευσης χρησιμοποιούμενων αρχείων, αποθήκευσης επιπλέον χαρακτηριστικών πεδίων στα αρχεία και υποστήριξης οποιασδήποτε υπηρεσίας ονομάτων για τον εντοπισμό των εξυπηρετητών. Όσων αφορά τις δυνατότητες που αφορούν την υλοποίηση του κοινού ονοματοχώρου, θεωρούμε σημαντικό πως υποστηρίζει την υπηρεσία αναφορικών υπερσυνδέσεων. Δυστυχώς, ο σχεδιασμός του CIFS έχει στοχευθεί σε γρήγορα τοπικά δίκτυα (LAN), με αποτέλεσμα η χρήση του σε δίκτυα μεγάλου εύρους, όπως WAN, να είναι ιδιαίτερα βλαβερή για την απόδοσή του, λόγω των μεγάλων συχνοτήτων απόκρισης, καθώς χρησιμοποιεί μεθόδους μαζικής εκπομπής για την αποστολή αιτημάτων και την εύρεση της

θέσης των πελατών στο δίκτυο. Επίσης, χαρακτηρίζεται από σημαντικά κενά ασφάλειας στην αρχιτεκτονική, τα οποία θα ήταν ανεπίτρεπτα σε μεγαλύτερη κλίμακα, από αυτήν ενός τοπικού δικτύου. Τέλος, σε πολλές υλοποιήσεις, η υπηρεσία αναπαραγωγής (*replication*) είναι περιορισμένη. Το CIFS, λοιπόν, κρίνεται μάλλον πρωτόγονο για χρήση σε ομοσπονδιακό σύστημα.

6.2 Andrew Filesystem

Αυτό το κατανεμημένο σύστημα αρχείων χαρακτηρίζεται από υψηλό επίπεδο ασφάλειας και δυνατότητες κλιμάκωσης (*scalability*) σε επίπεδο αριθμού χρηστών, κόμβων, καθώς και δεδομένων, ανεξάρτητα από την γεωγραφική τους θέση. Επίσης, υποστηρίζει προσωρινή τοπική αποθήκευση συχνά χρησιμοποιούμενων δεδομένων, καθώς και λειτουργίες αναπαραγωγής. Σημαντικό χαρακτηριστικό του AFS συστήματος αποτελούν τα *volumes*, τα οποία αποτελούν δένδρα υποκαταλόγων, αρχείων και σημείων προσάρτησης άλλων *volumes*, παρόμοια με τα σύνολα αρχείων στα ομοσπονδιακά συστήματα. Τα *volumes* δημιουργούνται από διαχειριστές, και συσχετίζονται με κάποια διαδρομή, δίνοντας την δυνατότητα στους αντίστοιχους χρήστες να κάνουν χρήση δεδομένων που αντιστοιχούν σε αυτά και να τα μεταβάλλουν. Όπως γίνεται αντιληπτό, τα χαρακτηριστικά αυτά ταυτίζονται με επιθυμητές ιδιότητες ομοσπονδιακών συστημάτων. Όμως, το AFS αδυνατεί να καλύψει την πιο σημαντική ανάγκη των ομοσπονδιακών συστημάτων αρχείων, καθώς απαιτεί ενσωμάτωση πολύπλοκου, ειδικά προσαρμοσμένου λογισμικού στον πυρήνα κάθε πελάτη και διαχειριστή του συστήματος. Αυτό αναιρεί την απαίτηση άγνοιας από τους πελάτες του συστήματος σε θέματα τοπικών ρυθμίσεων για την απόκτηση πρόσβασης στον ονοματοχώρο. Οι μηχανισμοί κλειδώματος που το χαρακτηρίζουν, επίσης, θεωρούνται τραχείς, καθώς αποκλείουν την πρόσβαση σε ολόκληρες μονάδες δεδομένων, και όχι απλά σε μεμονωμένα τμήματα αυτών που υπόκεινται σε αλλαγή. Έτσι, καθίσταται αδύνατη η πρόσθεση κοινόχρηστων πόρων όπως οι βάσεις δεδομένων, κάτι που δεν είναι επιθυμητό για ένα ομοσπονδιακό σύστημα. Παρόλα αυτά, λόγω του γεγονότος πως το AFS αποτελεί πρόγονο των ομοσπονδιακών συστημάτων, μπορούμε να πούμε ότι επηρέασε σε μεγάλο βαθμό την ανάπτυξή τους, δανείζοντας τους σημαντικά χαρακτηριστικά.

6.3 Plan 9

Το Plan 9 αποτελεί ένα κατανεμημένο λειτουργικό σύστημα, το οποίο χρησιμοποιεί τον ονοματοχώρο του συστήματος αρχείων για να χειριστεί σχεδόν κάθε πόρο του συστήματος. Το πρότυπο πρωτόκολλο ανταλλαγής μηνυμάτων 9P στο οποίο στηρίζεται το σύστημα, παρέχει ομοιόμορφο τρόπο πρόσβασης για όλους αυτούς τους πόρους. Τα χαρακτηριστικά

που μας οδηγούν στην εξέταση του Plan 9 ως βοηθητικού εργαλείου για την υλοποίηση ενός ομοσπονδιακού συστήματος είναι η αυξημένη έμφυτη ασφάλεια του, αλλά και η ευκολία δημιουργίας ενός ομοιόμορφου, και κοινόχρηστου ονοματοχώρου, η πρόσβαση του οποίου μπορεί να γίνεται διαφανώς από χρήστες. Το πρωτόκολλο 9P μπορεί εύκολα να λειτουργήσει σε ασφαλές περιβάλλον και να προσφέρει διαδικτυακή και διαδιεργασιακή επικοινωνία. Κάθε διεργασία έχει τον προσωπικό της ονοματοχώρο, όπου κρατά όλους τους πόρους στους οποίους έχει πρόσβαση. Οι πόροι αυτοί μπορούν να ανταλλαχθούν μεταξύ των διεργασιών και του δικτύου, ώστε να επιτραπεί διαμοιρασμός και ανεξαρτησία τοποθεσίας κατά την πρόσβαση. Η αρχιτεκτονική του συστήματος είναι προσανατολισμένη στον ονοματοχώρο, και δείχνει την δύναμη που παρέχει η καθολική ονομασία και οι μηχανισμοί πρόσβασης που αυτή συνεπάγεται. Ο προσωπικός ονοματοχώρος κάθε διεργασίας μπορεί να της εξασφαλίζει ασφάλεια και αυτονομία. Επειδή ονοματοχώροι περιορισμένης πρόσβασης θα είχαν μειωμένη χρησιμότητα, γίνεται χρήση μηχανισμού παραχώρησης εξουσιοδοτήσεων όταν προσαρτώνται απομακρυσμένα συστήματα αρχείων στο σύστημα, ώστε να επιτευχθεί καθολική ομοιομορφία. Παρόλα αυτά είναι προφανές, ότι επειδή το Plan 9 αποτελεί ένα ολοκληρωμένο λειτουργικό σύστημα, η εύρυθμη λειτουργία του σε ευρύτερα πλαίσια στηρίζεται σε ομοιογενείς κόμβους, και ένα κοινό πρωτόκολλο συστημάτων αρχείων. Ωστόσο, οι ιδέες και η εμπειρία που προσφέρει είναι κοινές με τους στόχους της προσπάθειας που περιγράφεται εδώ, και η υλοποίηση του, όπως και του AFS, έχει προσφέρει σημαντική γνώση όσων αφορά τα ομοσπονδιακά συστήματα αρχείων.

6.4 Glamour

Το Glamour είναι ένα ομοσπονδιακό σύστημα αρχείων, το οποίο σχεδιάστηκε το 2005, με σκοπό να επιτρέπει στους πελάτες να έχουν ομαλή πρόσβαση σε δεδομένα που βρίσκονται σε πολλαπλά ετερογενή συστήματα, ενδεχομένως απομακρυσμένα μεταξύ τους. Δεν αποτελεί υλοποίηση ευρύτερου κατανεμημένου συστήματος, αλλά επιτρέπει σε σύνολα ασθενώς συνδεδεμένων εξυπηρετητών να συμπεριφέρονται ως ένα σύστημα. Παρέχει λειτουργίες που είναι απαιτούμενες για ένα ομοσπονδιακό σύστημα, όπως: δημιουργία αντιγράφων σκιάς (κατά τη λειτουργία του εξυπηρετητή), μετακίνηση δεδομένων και προσωρινή αποθήκευση συχνά χρησιμοποιούμενων δεδομένων σε επιμέρους εξυπηρετητές. Λειτουργεί σε επίπεδο ενδιάμεσου λογισμικού, το οποίο μπορεί να συνυπάρξει με ήδη υπάρχοντα συστήματα και πρωτόκολλα παρέχοντας τις λειτουργίες του μέσω του NFSv4, χωρίς να απαιτεί ρύθμιση (τουλάχιστον δυναμική), από την πλευρά του πελάτη. Επίσης, δεν αποτελεί ένα νέο κοινό σύστημα αρχείων, αλλά αγνοεί τους τύπους των υπαρχόντων και λειτουργεί σε ανώτερο επίπεδο χωρίς να εξαρτάται από αυτά. Παρόλα αυτά, υπάρχουν και αρνητικά στοιχεία που στιγματίζουν την υλοποίηση αυτή. Ένα από αυτά ο αρχιτεκτονικός του σχεδιασμός, ο οποίος

στηρίζει την λειτουργία του συστήματος σε έναν κεντρικό εξυπηρετητή που προσφέρει τις σχετικές λειτουργίες πρόσβασης σε πελάτες, επιτελώντας ταυτόχρονα τις απαραίτητες εργασίες στα δεδομένα. Γενικότερα, η αρχιτεκτονική που χρησιμοποιείται διαχωρίζει το σύστημα σε μονάδες (*cells*), παρόμοιες με τα σύνολα αρχείων της υλοποίησης μας, οι οποίες χρησιμεύουν και ως διαχειριστικά πεδία, καθώς κάθε ένα από αυτά μπορεί να υπάγεται μόνο σε έναν διαχειριστή, ο οποίος μπορεί να είναι κοινός για οσεσδήποτε από αυτές. Επίσης αρνητικό, είναι το γεγονός πως αδυνατεί να δημιουργήσει ένα κοινό ονοματοχώρο, ο οποίος διευρύνεται σε παγκόσμιο επίπεδο (όπως πχ. ένα URL), εντούτοις έγιναν προσπάθειες προς την κατεύθυνση αυτή, με σκέψεις για χρήση του DNS πρωτοκόλλου στα υψηλότερα επίπεδα της ιεραρχίας της δενδρικής δομής, οι οποίες όμως δυστυχώς σταμάτησαν όταν το project τερματίστηκε. Σε γενικές γραμμές το Glamour παρουσιάζει πολλά κοινά στοιχεία με το AFS, όπως η αρχιτεκτονική διακριτών μονάδων (*cells*), η χρήση συνόλων δεδομένων (*filesets*) τα οποία συνδέονται με μια η περισσότερες τοποθεσίες (*locations*), χωρίς όμως να μπορεί να ενσωματώσει όλα τα πλεονεκτήματά του. Η κύρια διαφορά, όμως, έγκειται στη χρήση του NFSv4, το οποίο μπορεί να προσφέρει επιπλέον λειτουργίες από το AFS, όπως είναι οι προσωρινοί δείκτες αρχείων (*volatile filehandles*), οι παραπομπές (*client redirections*), καθώς και η αναπαραγωγή και μετακίνηση πηγών δεδομένων (*replication, migration*).

6.5 Storage Resource Broker

Ο Storage Resource Broker είναι ενδιάμεσο λογισμικό μεταξύ πελάτη και εξυπηρετητή, το οποίο μπορεί να προσφέρει ένα ομοιόμορφο περιβάλλον, όπου συνδυάζονται ετερογενείς πηγές δεδομένων οι οποίες γίνονται διαθέσιμες μέσω δικτύου, καθώς και πηγές δεδομένων που έχουν προκύψει ως αντίγραφα υπαρχόντων (*replication*). Έτσι, χαρακτηρίζεται από πληθώρα λειτουργιών, που το καθιστούν υποψήφιο για την υλοποίηση ενός ομοσπονδιακού συστήματος αρχείων. Επίσης, προσφέρει λύση κατανεμημένης διαχείρισης, υποστηρίζοντας συλλογικό, ελεγχόμενο διαμοιρασμό των δεδομένων. Αποτελείται από ένα πλούσιο σύνολο προγραμματιστικών διεπιφανειών (APIs), που επιτρέπουν χειρισμό των λειτουργιών του από εφαρμογές που λειτουργούν σε υψηλότερα επίπεδα. Το SRB, σε συνδυασμό με μετά-δεδομένα, μπορεί να δημιουργήσει ένα τρόπο πρόσβασης στα δεδομένα με βάση τα χαρακτηριστικά τους, αντί των φυσικών τοποθεσιών τους στον δίσκο. Αυτό μπορεί να επιτευχθεί με τη δημιουργία ενός λογικού ονοματοχώρου, ο οποίος αντιστοιχίζει για όλα τα αρχεία που τον αποτελούν το λογικό τους όνομα, με το αντίστοιχο φυσικό τους όνομα και την διαδρομή τους στο δίσκο. Στο SRB, τα αρχεία και τα αντικείμενα παρουσιάζονται ως σύνολα δεδομένων (ή οντότητες). Μέσω αυτών, τα δεδομένα μπορούν να συνταχθούν σε ιεραρχική δομή, το οποίο είναι και το επιθυμητό. Παρόλο που το SRB προσφέρει μια ολοκληρωμένη λύση για κατανεμημένη διαχείριση των δεδομένων που προσεγγίζει τις απαιτήσεις ενός

ομοσπονδιακού συστήματος, η εγκατάσταση και η προσαρμογή του είναι ιδιαίτερα επίπονες. Δεν αποτελεί έναν απλό ονοματοχώρο, και έτσι απαιτεί σημαντική υποδομή, ώστε να αποθηκεύσει και να προσφέρει πρόσβαση τελικά σε δεδομένα. Ακόμα πιο σημαντικό, όμως, είναι το γεγονός πως τα διαμοιραζόμενα δεδομένα που αντιπροσωπεύει αποθηκεύονται πρακτικά σε τοπικό επίπεδο στον εξυπηρετητή. Αυτό αντιτίθεται στην απαίτηση των ομοσπονδιακών συστημάτων που περιορίζει τον εξυπηρετητή στην υπόδειξη της πραγματικής θέσης των δεδομένων στον πελάτη, χωρίς να μεσολαβεί περαιτέρω. Στο SRB, η ενημέρωση των δεδομένων γίνεται μέσω συγχρονισμού, κάτι που μας οδηγεί στο συμπέρασμα πως χρήση του για τη δημιουργία ενός ομοσπονδιακού συστήματος, θα οδηγούσε σε μεγάλο όγκο μεταφερόμενων δεδομένων, καθώς και μεγάλο φόρτο στον κεντρικό εξυπηρετητή.

6.6 Domain Name System

Το DNS, μια ιεραρχική υπηρεσία ονομασίας για υπολογιστές, υπηρεσίες ή γενικότερα πόρους που είναι διαθέσιμοι μέσω του διαδικτύου, αναλαμβάνει την αντιστοίχιση διαφόρων πληροφοριών με ονομασίες δικτύου που τους έχουν ανατεθεί. Έχει μακρόχρονη ιστορία ευρωστίας σε περιβάλλοντα μεγάλης κλίμακας και ευρείας χρήσης, καλύπτοντας μια από τις βασικότερες απαιτήσεις ενός ομοσπονδιακού συστήματος. Η αρχιτεκτονική, αλλά και οι υπάρχουσες υλοποιήσεις ενδεχομένως μπορούν να χρησιμοποιηθούν για την αποθήκευση των ζητούμενων δεδομένων, και πιο ειδικά για την ονομασία των οργανισμών κοντά στην ρίζα του ονοματοχώρου, και συγκεκριμένα στο ριζικό σύνολο αρχείων. Η υπάρχουσα υποδομή για την διαχείριση πληροφοριών ονομασιών δικτύων, όμως, είναι μάλλον ακατάλληλη για να διαχειριστεί τα κατώτερα στρώματα εγγραφών. Αυτό οδηγεί στην ανάγκη χρήσης περισσότερων της μιας βάσης για την κάλυψη του ονοματοχώρου, καθώς η χρήση αποκλειστικά μιας βάσης για δύο τόσο ανόμοιους σκοπούς, θα απαιτούσε συντονισμό, ο οποίος θα οδηγούσε πιθανότατα σε περισσότερα προβλήματα, παρά πλεονεκτήματα. Παρόλα αυτά, εμμένουμε στην επιλογή του LDAP ως αποκλειστικού πρωτοκόλλου αντιστοίχισης, καθώς θεωρούμε ότι παρέχει μεγαλύτερη ελευθερία στην επιλογή των χαρακτηριστικών κάθε εγγραφής, καθώς και αφαίρεση ενδεχόμενης πολυπλοκότητας που θα προκαλούσε πιθανός αναγκαίος συγχρονισμός των δύο πρωτοκόλλων.

6.7 Global Namespace for files

Στην εργασία [Anderson00] γίνεται η αναλυτική θεωρητική προσέγγιση μιας υπηρεσίας (GNS) που προσφέρει δυνατότητες για τη σύνδεση ανομοιογενών πηγών δεδομένων. Ο όρος *"υπηρεσία"* χρησιμοποιείται με χαλαρότητα, για να υποδηλώσει συλλογή δεδομένων που σχετίζονται με ένα ονοματοχώρο, ο οποίος προσφέρει στους χρήστες του πρόσβαση σε

δεδομένα με ομοιογενή τρόπο. Το GNS διαθέτει υποσυστήματα που εξασφαλίζουν μια υποδομή ασφάλειας αλλά και υπηρεσίες διαχείρισης και εντοπισμού των δεδομένων και των θέσεων των χρηστών, με απώτερο στόχο την δημιουργία ενός ενιαίου ονοματοχώρου. Μέσω της εργασίας αυτής, μπορούμε να επιχειρήσουμε μια γενική θεωρητική προσέγγιση στο ζήτημα των ομοσπονδιακών συστημάτων, αφού εκτός από πρωτόκολλα κατανεμημένης πρόσβασης σε αρχεία, και τρόπους αντιστοίχισης πελατών με εξυπηρετητές, ο ονοματοχώρος όπως περιγράφεται από αυτή, καθορίζει και ένα πλέγμα συστημάτων αρχείων. Με τον όρο αυτό ορίζεται μια περίπτωση πλέγματος, η οποία ενοποιεί εξαγόμενα συστήματα αρχείων σε ένα σύστημα αρχείων, το οποίο είναι εικονικό και ενιαίο. Η περιγραφή του ονοματοχώρου GNS αναλύεται βαθύτερα στην εργασία [Anderson00], χωρίς όμως να ξεφεύγει ιδιαίτερα από τα θεωρητικά όρια.

6.8 Federated Filesystem

Στα προσχέδια [FedFS-Reqs],[FedFS-NSDB],[FedFS-Admin],[FedFS-DNS] περιγράφεται το FedFS, ένα πρωτόκολλο που μπορεί να χρησιμοποιηθεί για την δημιουργία ενός ομοσπονδιακού συστήματος. Το πρωτόκολλο αυτό, βρίσκεται ακόμα στο στάδιο της ανάπτυξης, και ξεκίνησε από μια ανεξάρτητη ερευνητική ομάδα, η οποία τον Οκτώβριο του 2008 εντάχθηκε στην ερευνητική ομάδα του NFSv4 της IETF. Το FedFS ορίζεται ως πρωτόκολλο που δίνει δυνατότητες διάσχισης ονοματοχώρων, καθώς και πρόσβασης σε δεδομένα που εκτείνονται μεταξύ συλλογών εξυπηρετητών, η διαχείριση των οποίων μπορεί να γίνεται από τους ίδιους, ή διαφορετικούς διαχειριστές. Το πρωτόκολλο περιγράφει αναλυτικά ένα σύνολο διεπαφών, μέσω της χρήσης των οποίων μπορούν να συνενωθούν εξυπηρετητές αρχείων και διαχειριστές σε μια ομοσπονδία εξυπηρετητών, που προσφέρει έναν ονοματοχώρο αποτελούμενο από συστήματα αρχείων, τα οποία βρίσκονται ή και εξάγονται από τους προαναφερθέντες εξυπηρετητές. Από τα προσχέδια του, αντλούμε χρήσιμες ιδέες, αλλά και αυστηρές αρχιτεκτονικές απαιτήσεις, που δίνουν μορφή στο ομοσπονδιακό σύστημα. Πολλές από αυτές μάλιστα, και ειδικότερα όσες έχουν χρησιμοποιηθεί και σε παρελθοντικές υλοποιήσεις, έχουν χρησιμοποιηθεί και στην υλοποίηση που περιγράφηκε. Το πρωτόκολλο μπορούμε να πούμε ότι εστιάζει στην εισαγωγή μηχανισμών που μπορούν να θέσουν σε συνεργασία διαφορετικά πρωτόκολλα εξαγωγής δεδομένων, παρόλο που στην παρούσα φάση δίνεται βάση μόνο στο πρωτόκολλο NFSv4.

7 *Επίλογος*

"επίλογος", (ο) ουσ. *το τελευταίο μέρος λόγου*
ή κειμένου || *(μτφ.) επακολούθημα*

Το κεφάλαιο αυτό συνοψίζει τα συμπεράσματα που προέκυψαν από την εργασία, δίνοντας ένα γενικό πλάνο των συμπερασμάτων που προέκυψαν από τα προηγούμενα κεφάλαια.

7.1 Σύνοψη και συμπεράσματα

Όπως θα πρέπει να έγινε αντιληπτό, ο τομέας των ομοσπονδιακών συστημάτων διαθέτει μεγάλες δυνατότητες επέκτασης καθώς πολλά θέματα μένουν να εξεταστούν. Οι υλοποιήσεις που δημιουργήθηκαν είχαν ως σκοπό την εκτίμηση του προβλήματος δημιουργίας ενός κοινού και ενιαίου ονοματοχώρου, ο οποίος προσφέρει ένα επίπεδο αφαίρεσης μεταξύ της πραγματικής θέσης των δεδομένων και της λογικής τους θέσης που αντιλαμβάνονται οι πελάτες της ομοσπονδίας. Λόγω του γεγονότος πως η δημιουργία του ενιαίου ονοματοχώρου στηρίζεται σε στατικά αρχεία ρυθμίσεων που βρίσκονται σε όλη την ομοσπονδία, η εισαγωγή δυναμικών χαρακτηριστικών σε αυτά είδαμε ότι μπορεί να μετατρέψει την μεταβολή της σύνθεσης του ονοματοχώρου σε τετριμμένο πρόβλημα.

Όμως δεν θα μπορούσαμε να θεωρήσουμε ότι το πρόβλημα έχει εξεταστεί πλήρως χωρίς τα αποτελέσματα των πειραμάτων που παρατέθηκαν. Μέσω αυτών διαπιστώθηκε ότι ο φόρτος των εξυπηρετητών δεν ξεπερνά το 0.3% σε επεξεργαστική ισχύ για δύο πελάτες, με ελάχιστη αύξηση σε σύγκριση με τη χρήση ενός, δίνοντας έτσι σταθερή βάση για κλιμάκωση του συστήματος. Επίσης, οι καθυστερήσεις στην εκτέλεση των λειτουργιών για πολλά

σύνολα αρχείων, σε σχέση με την εκτέλεσή τους χωρίς την αντιστοίχιση (ή με την εκτέλεση για ένα σύνολο αρχείων που είναι παρόμοια περίπτωση) ανήκουν σε γενικές γραμμές στην τάξη των millisecond για τους αριθμούς συνόλων αρχείων που είδαμε. Με αυτά τα δεδομένα, μπορούμε να αποφανθούμε πλέον ότι η ενσωμάτωση του μηχανισμού αντιστοίχισης δεν δημιουργεί καθυστερήσεις που θα μπορούσαν να είναι απαγορευτικές για πραγματικά συστήματα, καθιστώντας τις υλοποιήσεις ως ένα χρήσιμο πρώτο βήμα στην προσέγγιση της εφαρμογής των ομοσπονδιακών συστημάτων.

7.2 Μελλοντικές επεκτάσεις

Σε ρεαλιστική βάση, η παρούσα εργασία άγγιξε μόνο την επιφάνεια του προβλήματος δημιουργίας ενός ομοσπονδιακού συστήματος. Ζητήματα όπως η εξέταση των υλοποιήσεων σε περιβάλλοντα με μεγάλη κλιμάκωση, η υλοποίηση υποστήριξης για περισσότερα περιβάλλοντα χρήσης ή πρωτόκολλα επικοινωνίας και κάποιο ενσωματωμένο σύστημα ασφάλειας απουσιάζουν για ευνόητους λόγους, όμως θεωρείται άκρως σημαντικό να γίνουν κινήσεις και προς αυτές τις κατευθύνσεις στο μέλλον.

Δυστυχώς, τα ομοσπονδιακά συστήματα βρίσκονται ακόμα σε πρωταρχικό στάδιο ανάπτυξης και αυτό δυσκόλεψε ιδιαίτερα την έρευνα που έγινε για την εργασία. Τη στιγμή που γράφονται αυτές οι γραμμές, γίνονται προσπάθειες από την ερευνητική ομάδα ανάπτυξης του NFSv4 της IETF να παραχθούν κάποια πρότυπα τα οποία θα περιγράφουν αναλυτικά την έννοια και τα συστατικά ενός ομοσπονδιακού συστήματος. Αυτό ίσως οδηγήσει την έννοια της ομοσπονδίας σε ταύτιση με το πρωτόκολλο, όμως σε κάθε περίπτωση μπορούμε να ελπίζουμε ότι θα είναι η μόνο η αρχή του ερευνητικού ενδιαφέροντος για τον τομέα. Σε αυτό το συμπέρασμα οδηγεί και το γεγονός πως ένας άλλος τομέας, το cloud computing, ταυτίζεται από πολλούς με τα ομοσπονδιακά συστήματα, καθώς τα χαρακτηριστικά τους σχετίζονται σε αφαιρετικό επίπεδο.

Σε κάθε περίπτωση, με την τεχνολογία που είναι διαθέσιμη σήμερα στους τομείς των δικτύων, αλλά και της αποθήκευσης δεδομένων, η παρούσα τεχνολογία εμφανίζεται πολλά υποσχόμενη, με πραγματικές εφαρμογές στην καθημερινότητα. Η δυνατότητα πρόσβασης σε απομακρυσμένα δεδομένα με τη χρήση δυναμικών τοποθεσιών θα μπορούσε, σε πολλές περιπτώσεις, να αντικαταστήσει την ανάγκη τοπικής αποθήκευσης, αλλά αυτό που θα ήταν πραγματικά καινοτόμο, είναι πως όλοι οι χρήστες θα μπορούσαν να έχουν πρόσβαση στα ίδια δεδομένα μέσω των ίδιων τοποθεσιών, δημιουργώντας έτσι ένα είδος URL στο επίπεδο των συστημάτων αρχείων, όπως ακριβώς χρησιμοποιείται και στο διαδίκτυο.

8 *Βιβλιογραφία*

[CITI0601] Jiaying Zhang, Peter Honeyman. *"Naming, Migration, and Replication for NFSv4",* CITI Tech. Report, University of Michigan, January 2006.

[Charbon06] Aurelien Charbon, Tony Reix, Vincent Roqueta, Bryce Harrington, Bruce J. Fields, Trond Myklebust, Suresh Jayaraman, Jeff Needle, Barry Marson. *"NFSv4 Test Project"*, Open Linux Symposium, August 2006, pp. 115-134.

[FedFS-Reqs] James Lentini, Craig Everhart, Daniel Ellard, Renu Tewari, Manoj Naik. *"Requirements for Federated File Systems"*, Informational Internet Draft, IETF NFSv4 Working Group, April 20, 2009.

[FedFS-NSDB] James Lentini, Craig Everhart, Daniel Ellard, Renu Tewari, Manoj Naik. *"NSDB Protocol for Federated File Systems"*, Standards Track Internet Draft, IETF NFSv4 Working Group, February 20, 2009.

[FedFS-Admin] James Lentini, Craig Everhart, Daniel Ellard, Renu Tewari, Manoj Naik. *"Administration Protocol for Federated File Systems"*, Standards Track Internet Draft, IETF NFSv4 Working Group, March 6, 2009.

[FedFS-DNS] Craig Everhart, Andy Adamson, Jiaying Zhang. *"Using DNS SRV to Specify a Global File Name Space with NFS Version 4"*, Standards Track Internet Draft, IETF NFSv4 Working Group, September 26, 2009.

[Anderson00] O. T. Anderson, L. Luan, C. Everhart, M. Pereira, R. Sarkar, J. Xu. *"Global Namespace for files"*, IBM Systems Journal, January 12, 2004.

[Pawlowski00] Brian Pawlowski, Spencer Shepler, Carl Beame, Brent Callaghan,

Michael Eisler, David Noveck, David Robinson, Robert Thurlow. *"The NFS Version 4 Protocol"*, In Proceedings of the 2nd International System Administration and Networking Conference (SANE '00), 2000.

[Glamour05] Renu Tewari, Jonathan M. Haswell, Manoj P. Naik, Steven M. Parkes. *"Glamour: A Wide-Area Filesystem Middleware Using NFSv4"*, IBM Research Report, Almaden Research Center, July 12, 2005.

[Satyanar90] Mahadev Satyanarayanan. *"Scalable, Secure, and Highly Available Distributed File Access"*, Computer, Carnegie Mellon University, May 1990.

[Birrell93] Andrew D. Birrell, Andy Hisgen, Chuck Jerian, Timothy Mann, Garret Swart. *"The Echo Distributed File System"*, DIGITAL Research Report, September 10, 1993.

[Douceur06] John R. Douceur, Jon Howell. *"Distributed Directory Service in the Farsite File System"*, In Proceedings of the 7th Symposium on Operating Systems Design and Implementation (OSDI '06), 2006.

[RFC4506] M. Eisler, Ed. *"XDR: External Data Representation Standard"*, Standards Track Internet Draft, IETF Network Working Group, May, 2006.

[RFC3530] S. Shepler, B. Callaghan, D. Robinson, R. Thurlow, C. Beame, M. Eisler, D. Noveck. *"Network File System (NFS) version 4 Protocol"*, Standards Track Internet Draft, IETF Network Working Group, April, 2003.

[Dike06] J. Dike. *User Mode Linux*, Bruce Perens' Open Source Series, Prentice Hall, April 22, 2006.

[Kernel29.4] The Linux Kernel Archives, kernel 2.6.29.4, stable. http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.29.4.tar.bz2